COMMUNICATIONS OF THE ACM
CACM.ACM.ORG

Moving Towards Reliable Autonomous Machines: The Vulnerability-Adaptive Protection Paradigm

Zishen Wan[1*], Yiming Gan[2*], Bo Yu[3], Shaoshan Liu[3], Arijit Raychowdhury[1], Yuhao Zhu[2]

[1]Georgia Institute of Technology [2]University of Rochester
[3]Shenzhen Institute of AI and Robotics for Society
*(*Equal Contributions)*

Research and Advances, Communication of the ACM

Georgia Tech
UNIVERSITY of ROCHESTER
AIRS

Outline

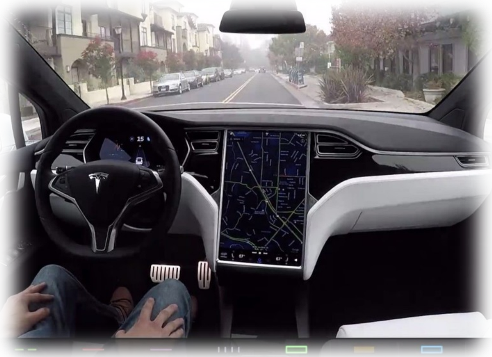
- Motivation – Why autonomous system needs reliability
- What is Autonomous Machine System
  - The concept of frontend and backend autonomous machine kernels
- VAP Framework
  - System performance and resiliency characterization
  - Vulnerability-adaptive protection
- Evaluations
  - Autonomous vehicle and drone

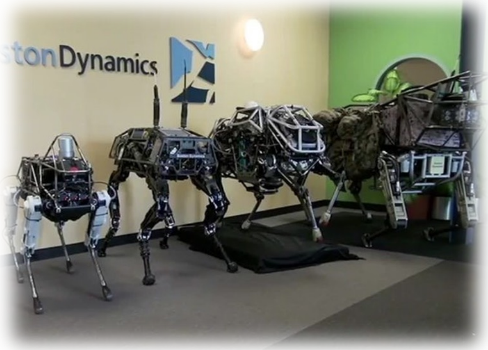
Motivation

Autonomous Machines

Motivation

Autonomous Machines





Performance

Goal: Improve task accuracy
(Autonomy Algorithms)

Efficiency

Goal: Improve data and compute efficiency
(Hardware Architecture)

Motivation

[1] Telsa Autopilot System Found Probably at Fault in 2018 Crash, The New York Times, 2021
[2] Surviving an In-Flight Anomaly: What Happened on Ingeuity's Sixth Flight, NASA Science, 2021

Motivation

Goal: Improve operational resiliency under faults without degrading performance and efficiency

Reliability

Autonomous Machines

Performance

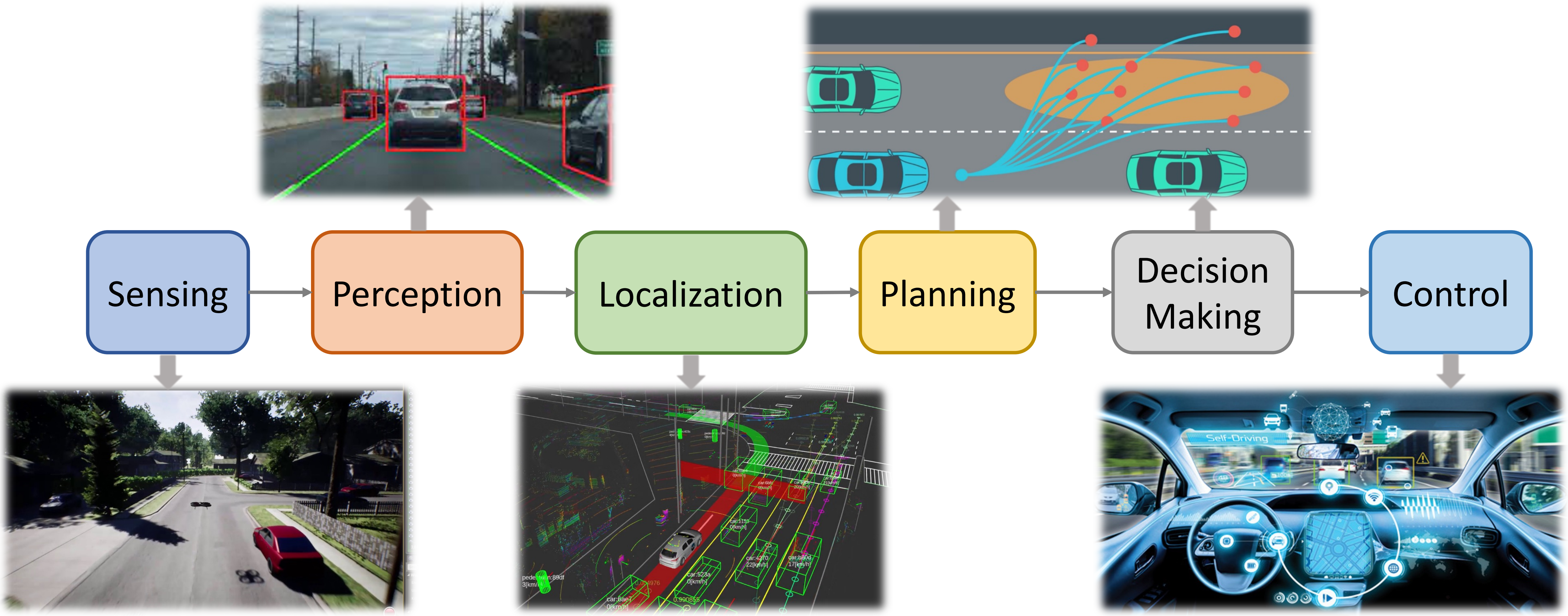
Goal: Improve task accuracy
(Autonomy Algorithms)

Performance-Efficiency-Reliability
Co-Optimization

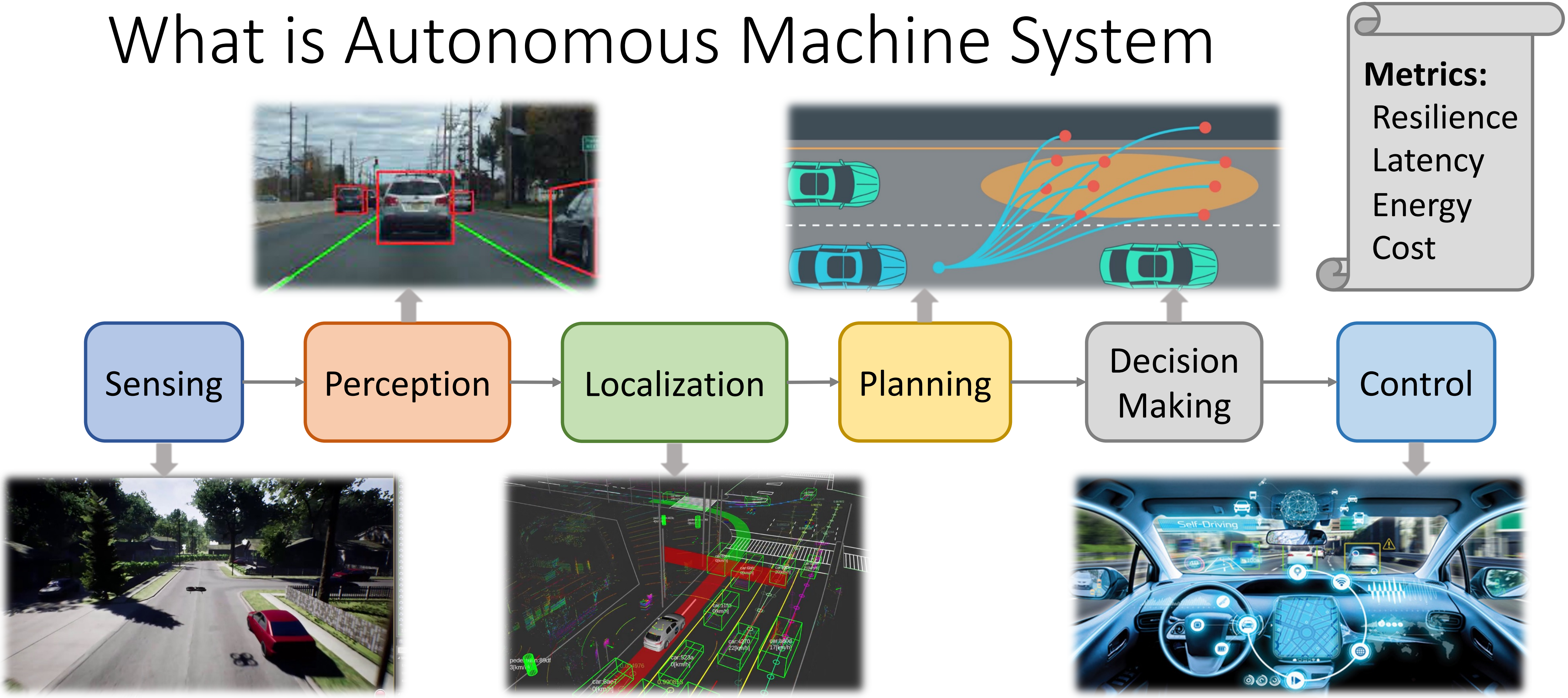
Efficiency

Goal: Improve data and compute efficiency
(Hardware Architecture)

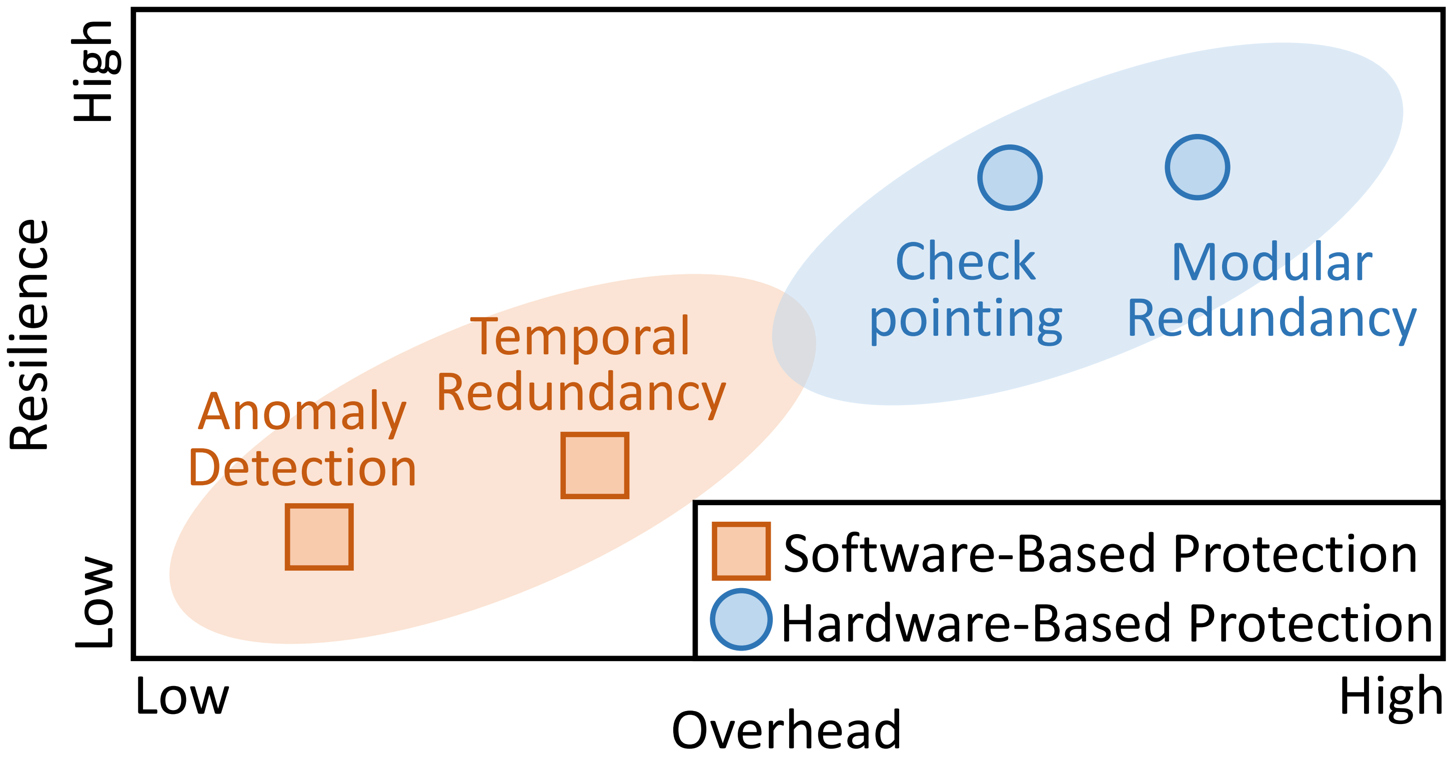
What is Autonomous Machine System



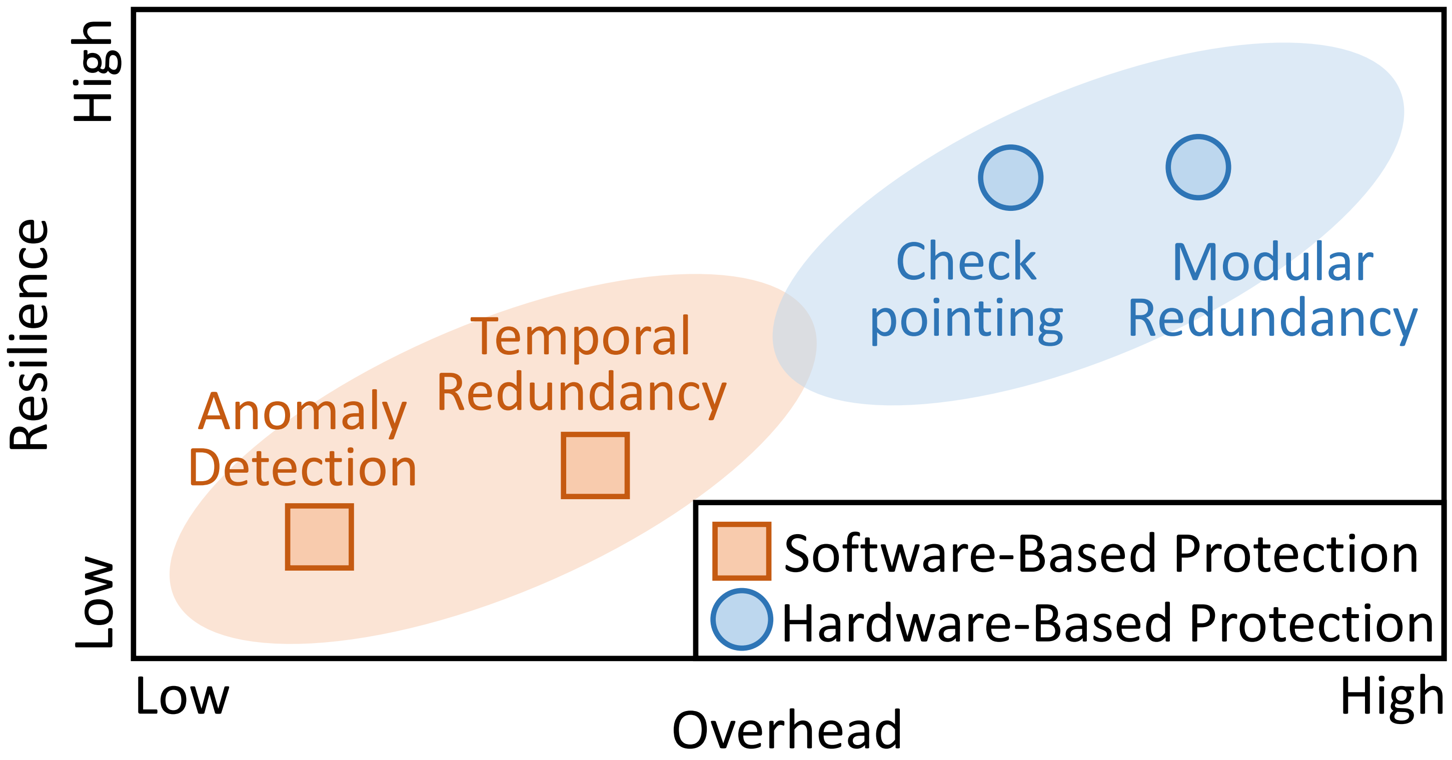
What is Autonomous Machine System



Design Landscape of Protection Techniques



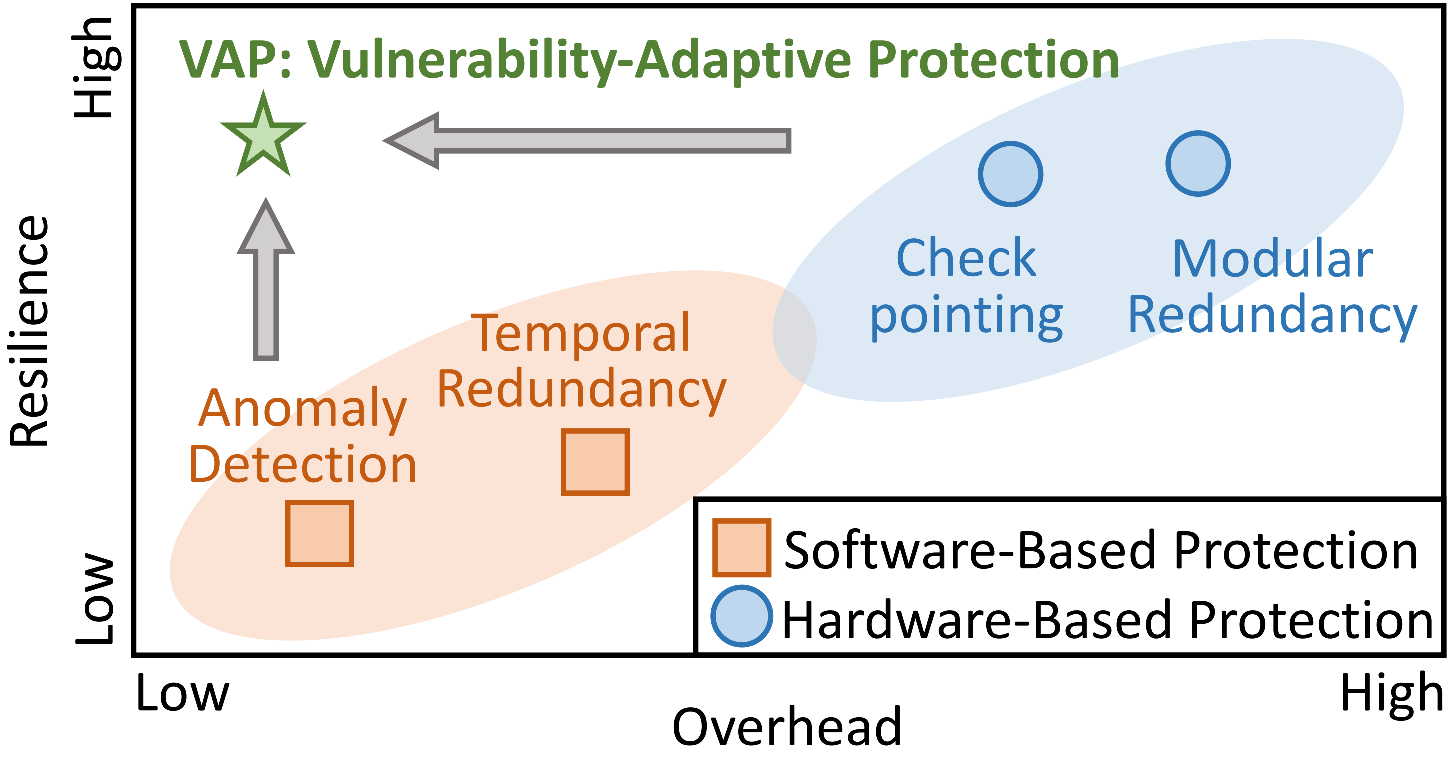
Challenge



<u>Challenge</u>: Today's resiliency solutions are of "*<u>one-size-fits-all</u>*" nature: they use the same protection scheme throughout entire autonomous machine, bringing *<u>trade-offs</u>* between resiliency and cost

How to provide high protection coverage
while introducing little cost
for autonomous machine system?

Insight & Solution



Insight & Solution: exploit the *inherent resiliency variations* in autonomous machine system to conduct *vulnerable-proportional protection (VPP)*

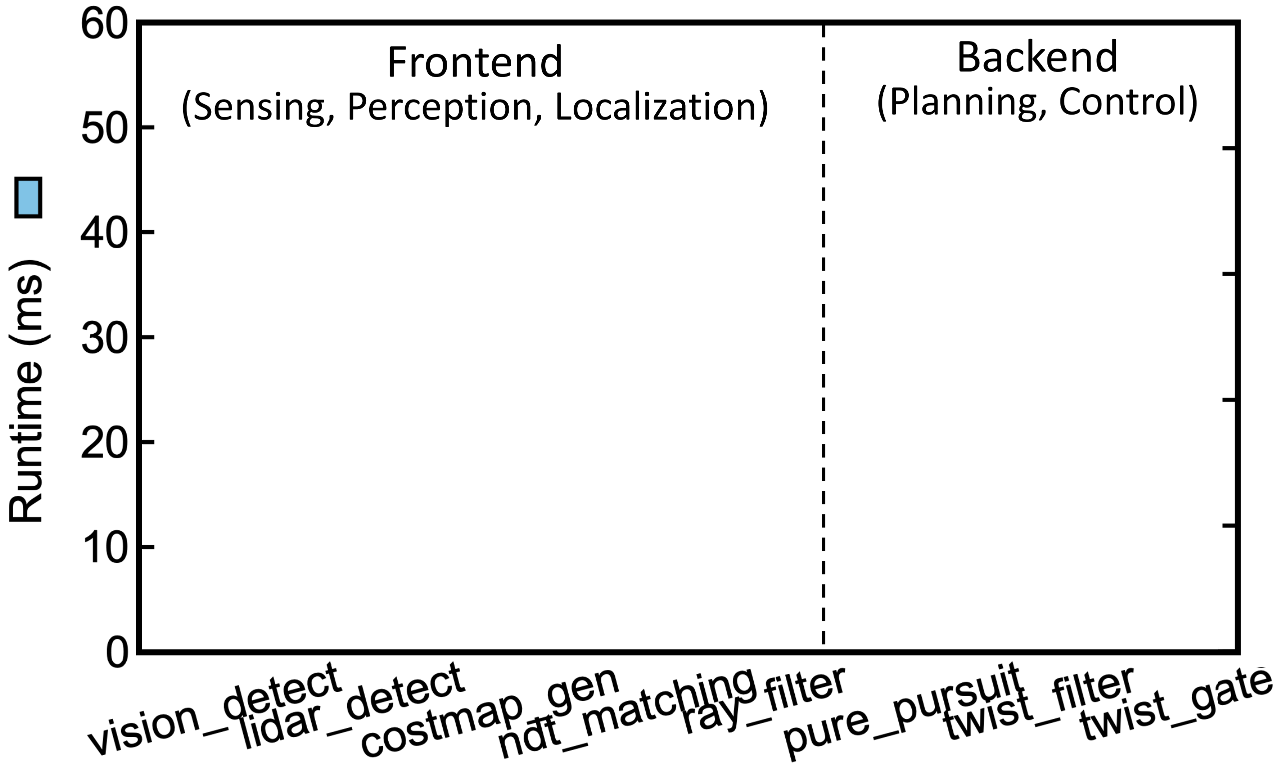
VAP Overview

(VPP: Vulnerability-Adaptive Protection)

VAP Overview

(VPP: Vulnerability-Adaptive Protection)

System Characterization - Autonomous Vehicle

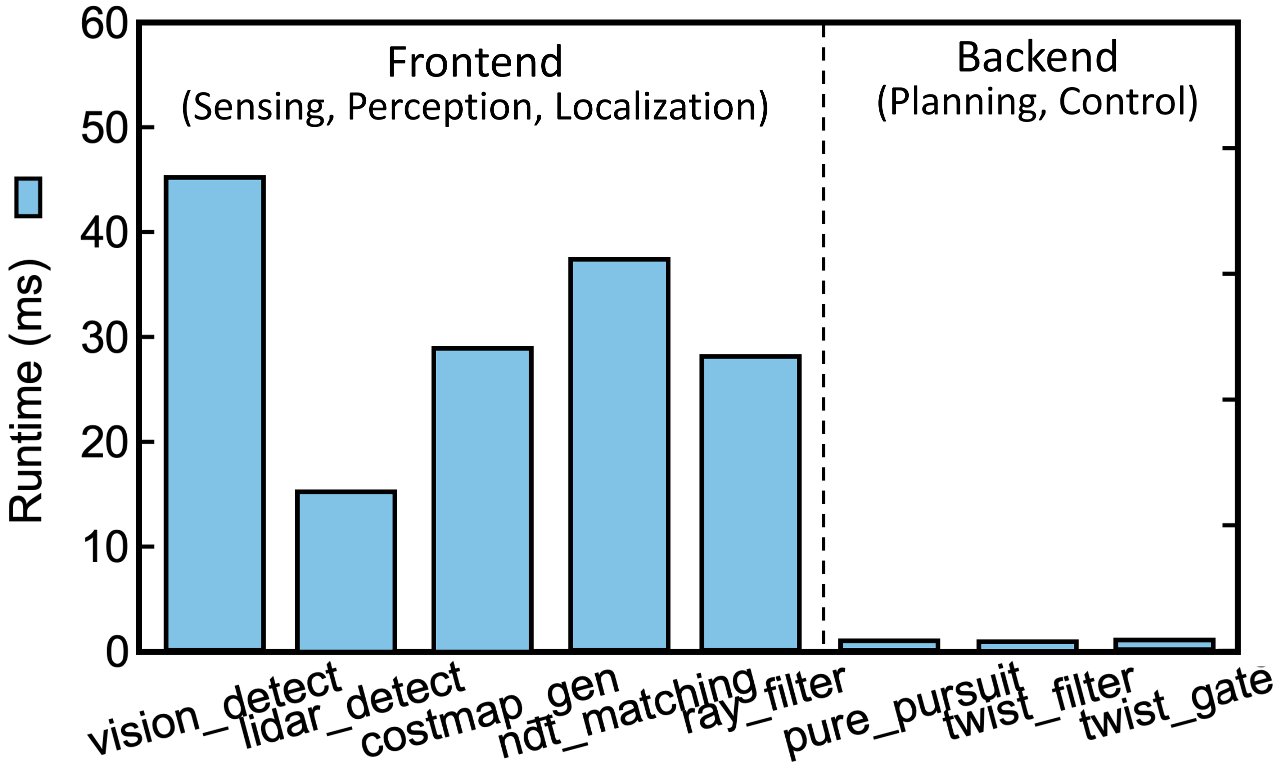


Experimental Setup

- Platform: Autonomous Vehicle (Autoware[1])

[1] Kato et al, IEEE Micro, 2015

System Characterization - Autonomous Vehicle



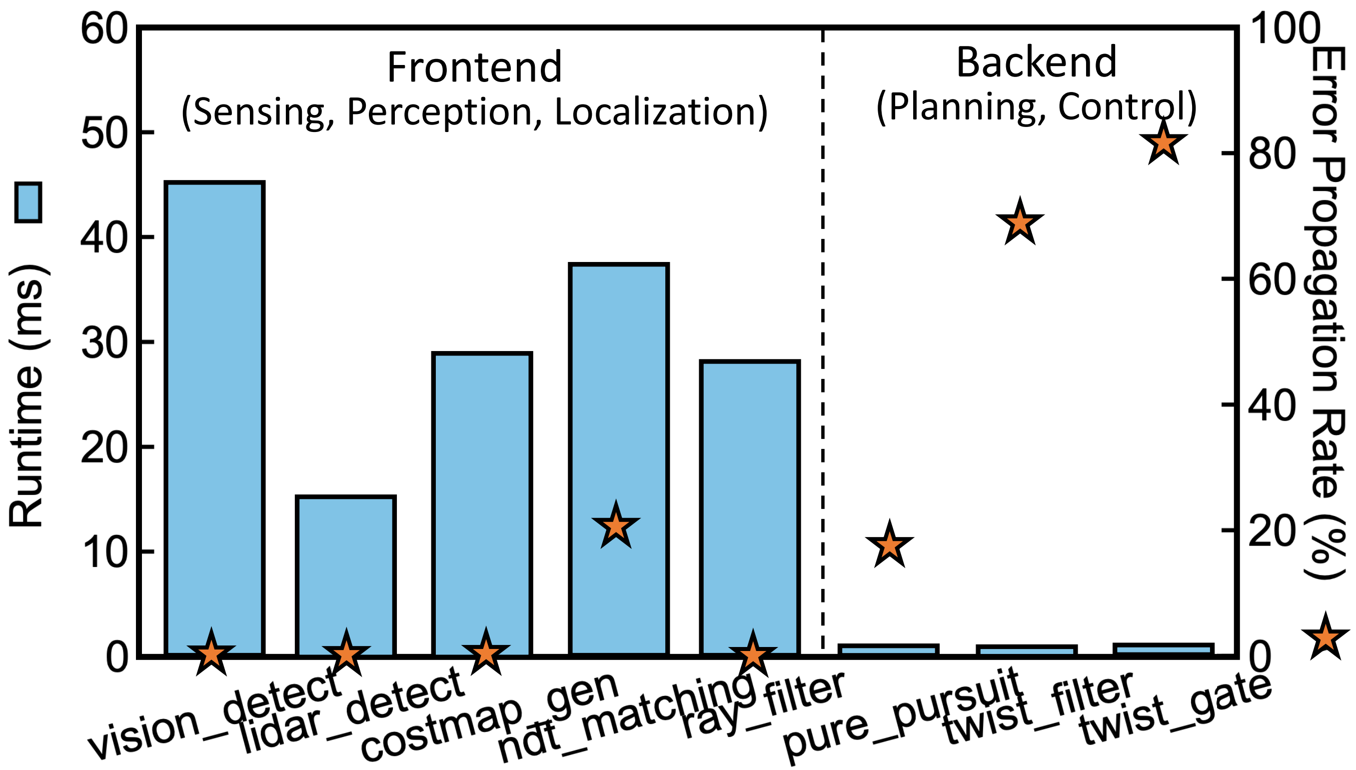
Experimental Setup

- Platform: Autonomous Vehicle (Autoware[1])

[1] Kato et al, IEEE Micro, 2015

Insight: frontend high latency
backend low latency

System Characterization - Autonomous Vehicle



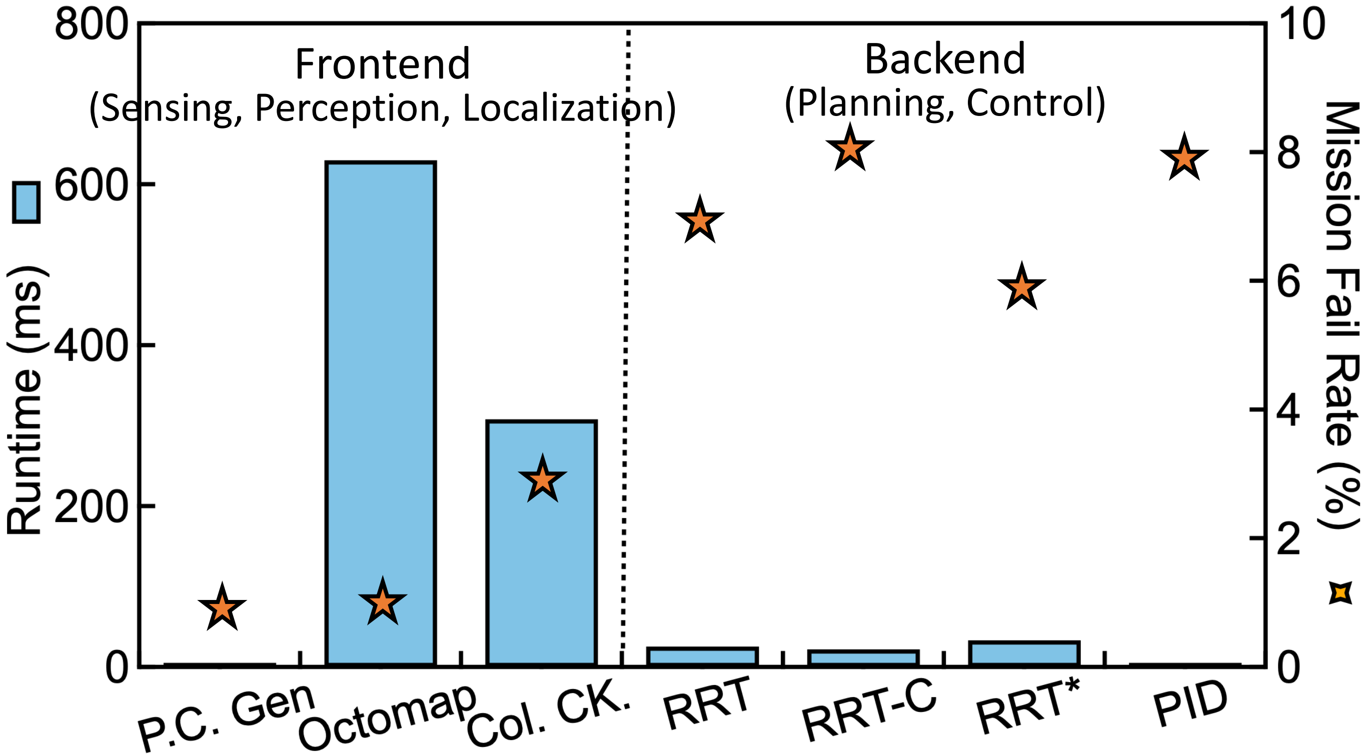
Experimental Setup

- Platform: Autonomous Vehicle (Autoware[1])
- Reliability: soft errors

[1] Kato et al, IEEE Micro, 2015

Insight: frontend high latency, low vulnerability
backend low latency, high vulnerability

System Characterization - Autonomous Drone



Experimental Setup

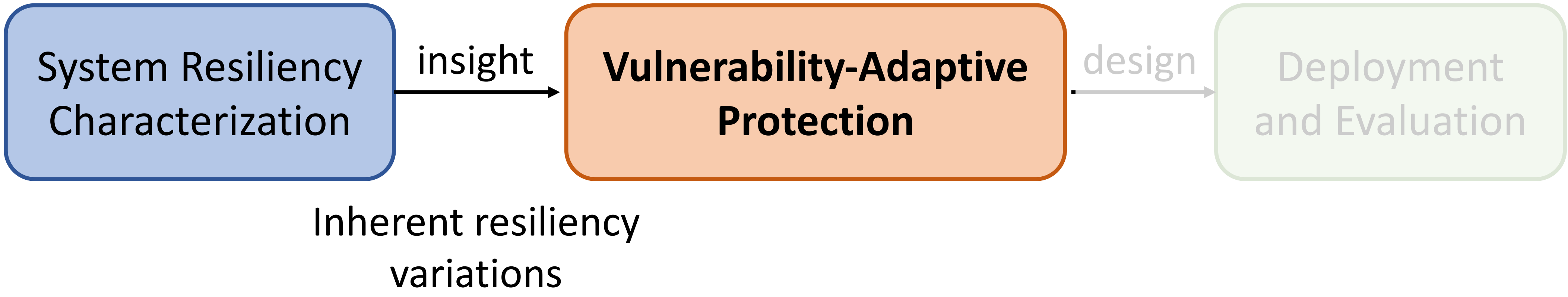
- Platform: Autonomous Drone (MAVBench[2])
- Reliability: soft errors

[2] Boroujerdian et al, MICRO, 2018

Insight: frontend high latency, low vulnerability
backend low latency, high vulnerability

VAP Overview

(VAP: Vulnerability-Adaptive Protection)

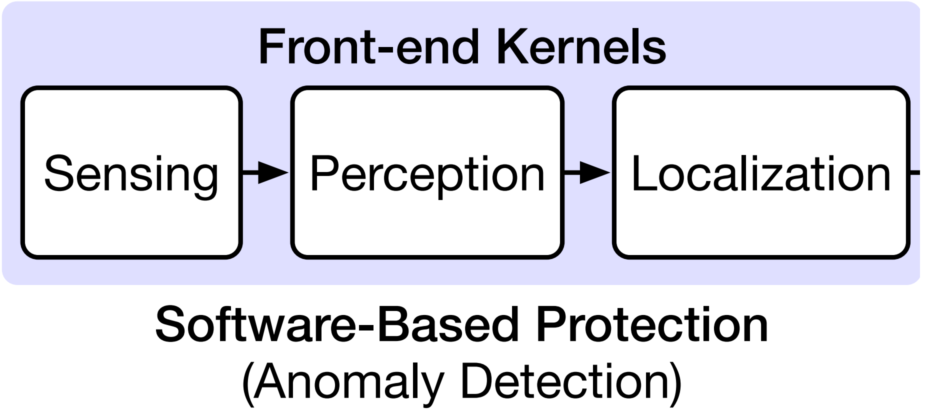


Vulnerability-Adaptive Protection

- **Design Principle**: the protection budget, be it spatially or temporally, should be allocated inversely proportionally to kernel inherent resilience

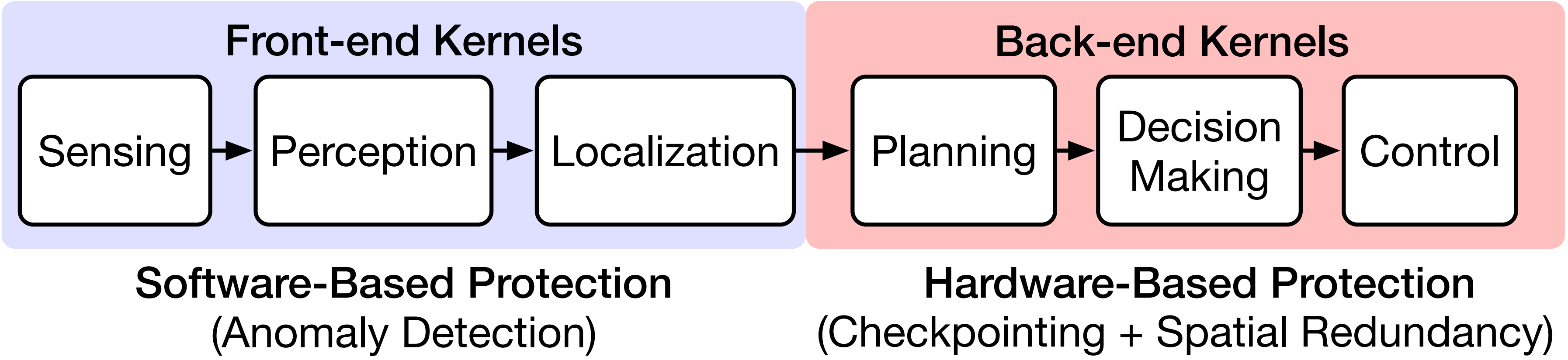
Vulnerability-Adaptive Protection

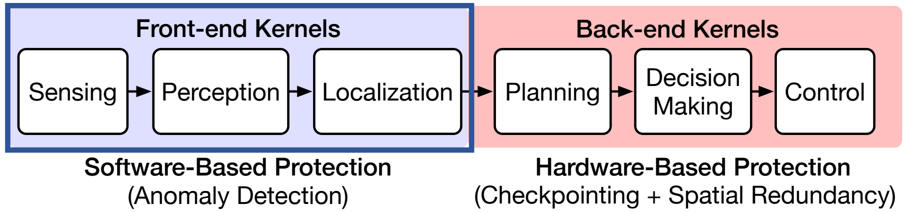
- **Design Principle**: the protection budget, be it spatially or temporally, should be allocated inversely proportionally to kernel inherent resilience
  - **Frontend**: low vulnerability -> lightweight software-based protection



Vulnerability-Adaptive Protection

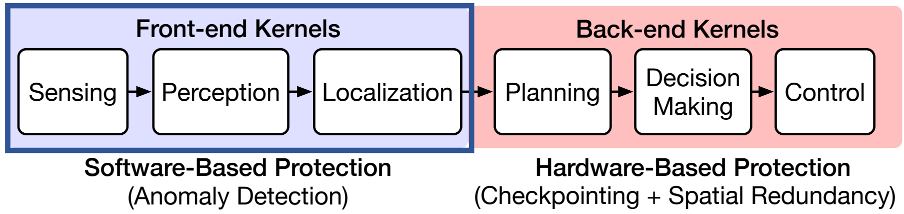
- **Design Principle**: the protection budget, be it spatially or temporally, should be allocated inversely proportionally to kernel inherent resilience
  - **Frontend**: low vulnerability -> lightweight software-based protection
  - **Backend**: high vulnerability -> more protection efforts, hardware-based protection





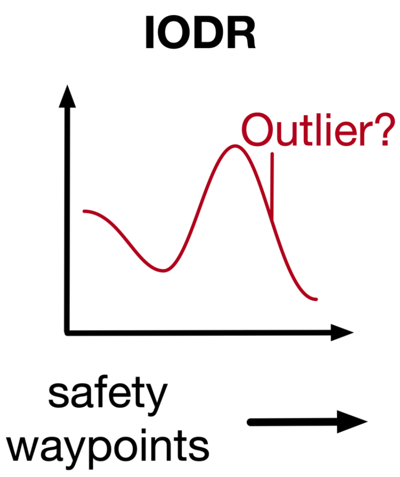
Frontend: Anomaly Detection

- **Frontend Insights**:
  - Strong **temporal consistency** of inputs and outputs
  - Inherent **error-masking** and error-attenuation capabilities
  - **Rare false positive** detection

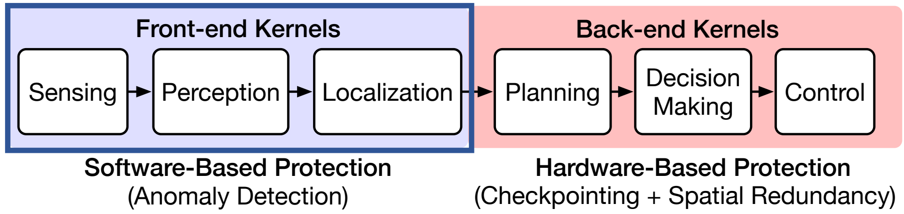


Frontend: Anomaly Detection

- **Frontend Insights**:
  - Strong **temporal consistency** of inputs and outputs
  - Inherent **error-masking** and error-attenuation capabilities
  - **Rare false positive** detection



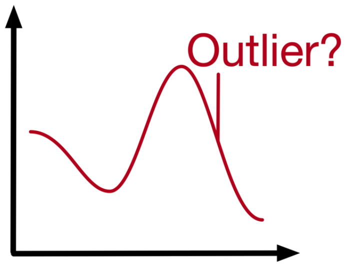
IODR: Input Outlier Detection and Resetting



Frontend: Anomaly Detection

- **Frontend Insights**:
  - Strong **temporal consistency** of inputs and outputs
  - Inherent **error-masking** and error-attenuation capabilities
  - **Rare false positive** detection

IODR

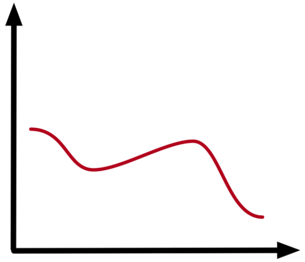


safety waypoints →

```
void ChangeWp(const VelocitySetInfo& vs_info, float
safety_wp):
{
   double deceleration = 0.0;
   double velocity_set =0.0;
   cond1 = detect(vs_info);
   if (cond1)
   {
       final_wp = change(safety_wp);
   }
   else
   {
       final_wp = change(safety_wp);
   }
}
```

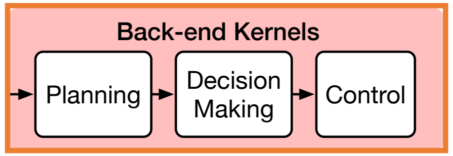
→ final waypoints

OODR



IODR: Input Outlier Detection and Resetting

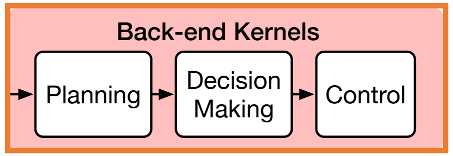
OODR: Output Outlier Detection and Resetting



Hardware-Based Protection
(Checkpointing + Spatial Redundancy)

Backend: Redundancy & Checkpointing

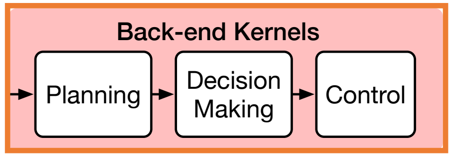
- **Backend Insights**:
  - **Critical** to errors
  - **Extremely lightweight** that do not involve complex computation
  - **More false positive** detection cases



Hardware-Based Protection
(Checkpointing + Spatial Redundancy)

Backend: Redundancy & Checkpointing

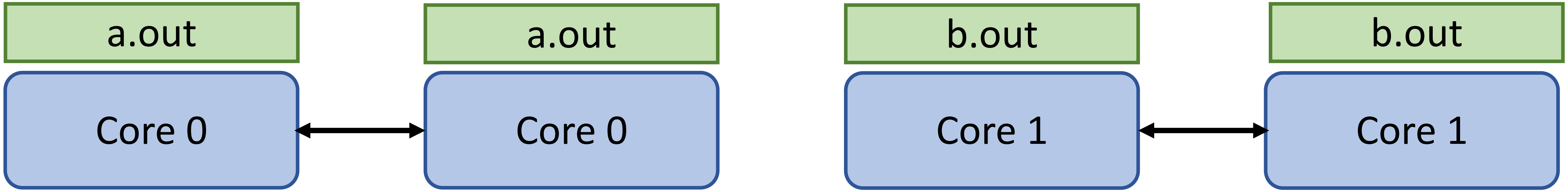
- **Backend Insights**:
  - **Critical** to errors
  - **Extremely lightweight** that do not involve complex computation
  - **More false positive** detection cases

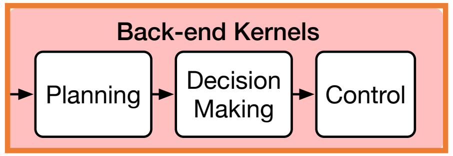


Hardware-Based Protection
(Checkpointing + Spatial Redundancy)

Backend: Redundancy & Checkpointing

- **Backend Insights**:
  - **Critical** to errors
  - **Extremely lightweight** that do not involve complex computation
  - **More false positive** detection cases

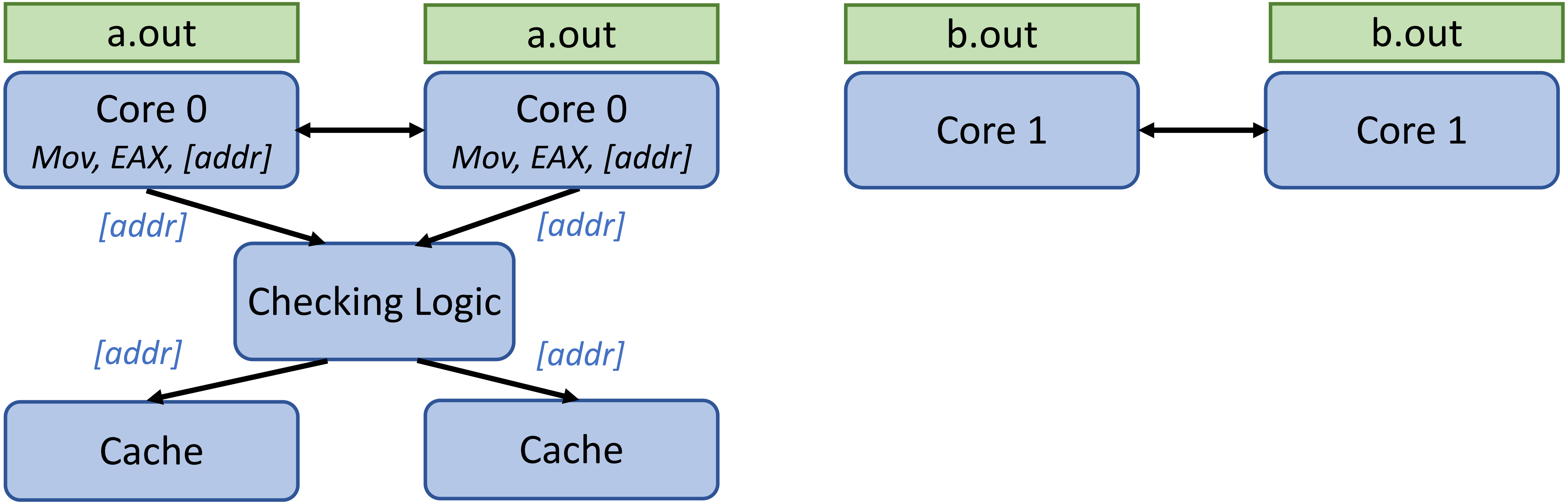


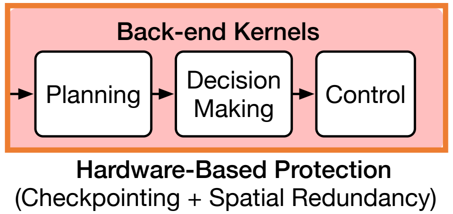


Hardware-Based Protection
(Checkpointing + Spatial Redundancy)

Backend: Redundancy & Checkpointing

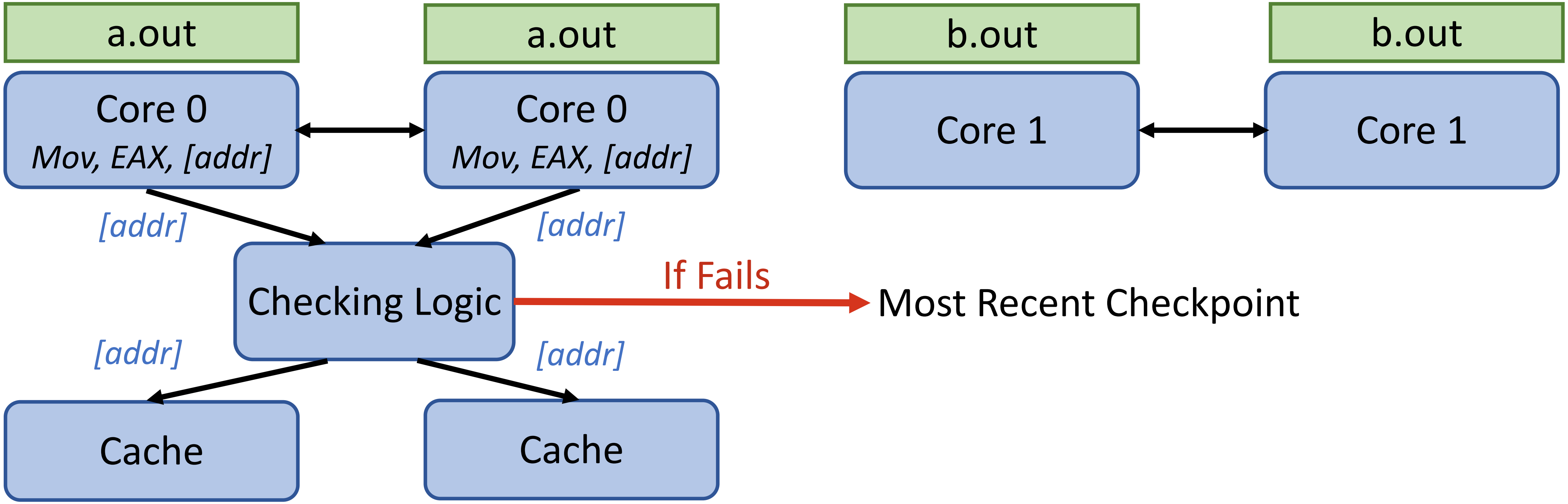
- **Backend Insights**:
  - **Critical** to errors
  - **Extremely lightweight** that do not involve complex computation
  - **More false positive** detection cases





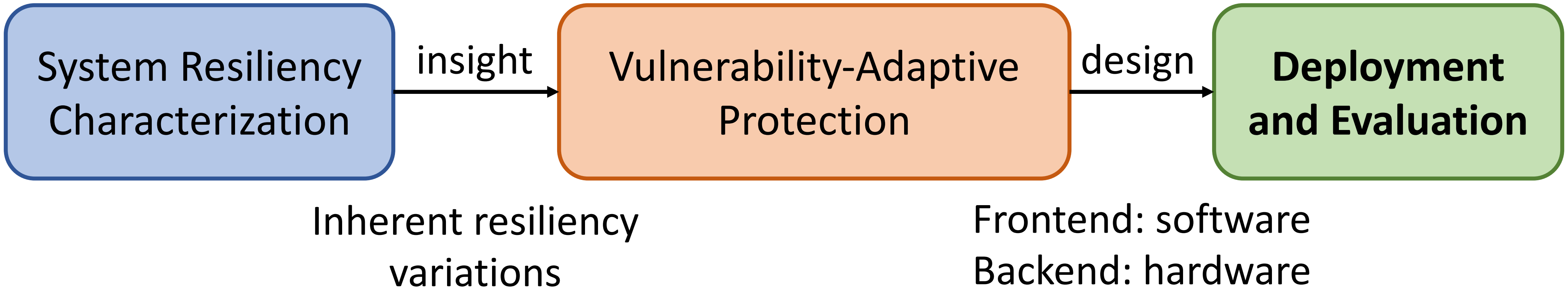
Backend: Redundancy & Checkpointing

- **Backend Insights**:
  - **Critical** to errors
  - **Extremely lightweight** that do not involve complex computation
  - **More false positive** detection cases



VAP Overview

(VAP: Vulnerability-Adaptive Protection)



Evaluation – Autonomous Vehicle

Fault Protection Scheme	
Baseline	No Protection
Software	Anomaly Detection
	Temporal Redundancy
Hardware	Modular Redundancy
	Checkpointing
Adaptive Protection Paradigm (*VPP*) Front-end Software + Back-end Hardware	

Experimental Setup

- Platform: Autonomous Vehicle (Autoware[1])

[1] Kato et al, IEEE Micro, 2015

Evaluation – Autonomous Vehicle

Fault Protection Scheme		Resilience: Error Propagation Rate (%)
Baseline	**No Protection**	46.5
Software	**Anomaly Detection**	24.2
	Temporal Redundancy	11.7
Hardware	**Modular Redundancy**	0
	Checkpointing	0
Adaptive Protection Paradigm (*VPP*) Front-end Software + Back-end Hardware		0

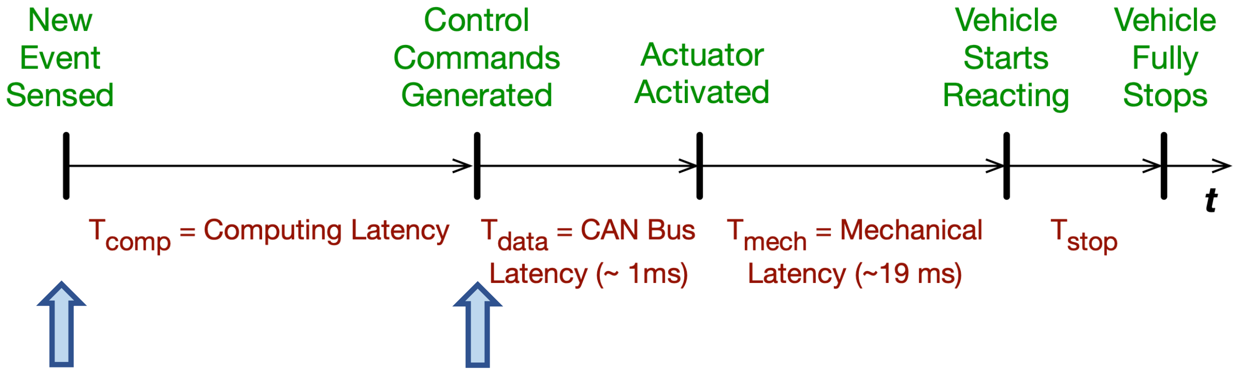
Experimental Setup
- Platform: Autonomous Vehicle (Autoware[1])
- Reliability: soft errors

[1] Kato et al, IEEE Micro, 2015

Takeaway: VPP *improves resilience* and *reduces error propagation rate* by (1) leveraging inherent error-masking capabilities of front-end and (2) strengthening back-end resilience by hardware-based redundancy and checkpointing.

Evaluation – Autonomous Vehicle

Fault Protection Scheme		Resilience	Latency and Object Distance
		Error Propagation Rate (%)	Compute Latency (*ms*)
Baseline	**No Protection**	46.5	164
Software	**Anomaly Detection**	24.2	245
	Temporal Redundancy	11.7	347
Hardware	**Modular Redundancy**	0	164
	Checkpointing	0	610
Adaptive Protection Paradigm (*VPP*) Front-end Software + Back-end Hardware		0	173

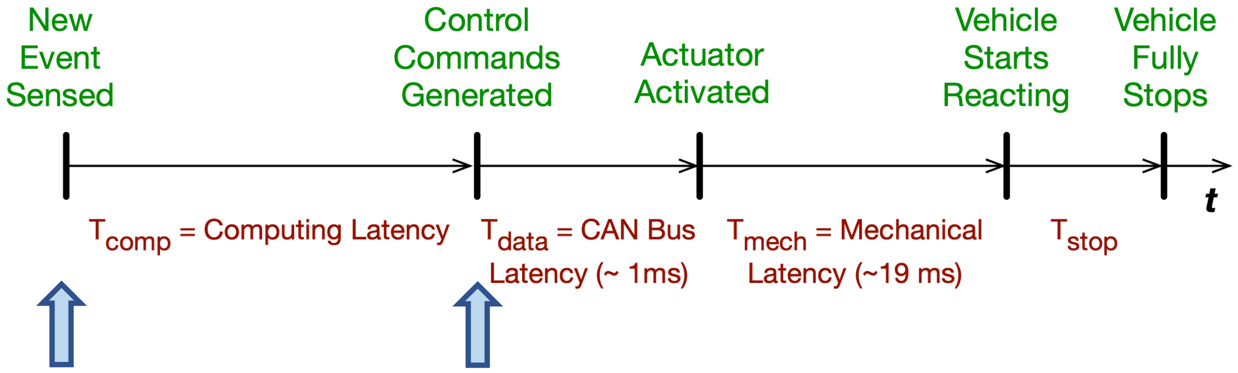


	Perception	Localization	Planning	Control	Total
No Protection	58	69	35	2	164
Anomaly Detection	64	72	106	3	245
Checkpointing	216	256	131	7	610
VAP	64	72	35	2	173

Compute latency breakdown of different protection schemes in the autonomous vehicle system

Evaluation – Autonomous Vehicle

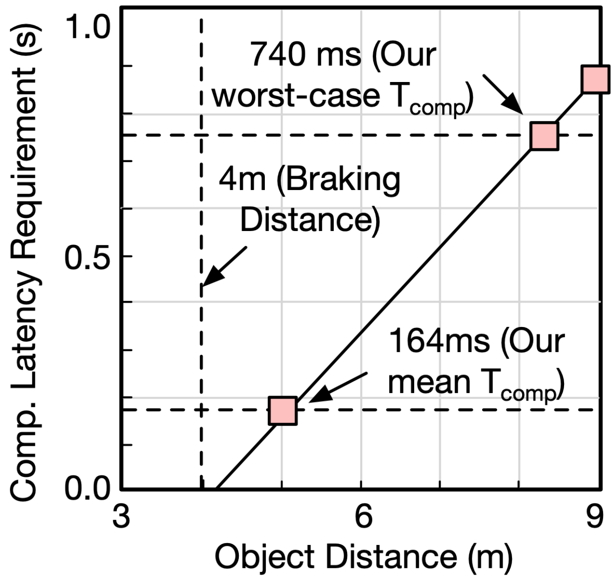
Fault Protection Scheme		Resilience	Latency and Object Distance
		Error Propagation Rate (%)	Compute Latency (ms)
Baseline	**No Protection**	46.5	164
Software	**Anomaly Detection**	24.2	245
	Temporal Redundancy	11.7	347
Hardware	**Modular Redundancy**	0	164
	Checkpointing	0	610
Adaptive Protection Paradigm (*VPP*) Front-end Software + Back-end Hardware		0	173



Takeaway: VPP reduce end-to-end compute latency overhead.

Evaluation – Autonomous Vehicle

Fault Protection Scheme		Resilience	Latency and Object Distance	
		Error Propagation Rate (%)	Compute Latency (*ms*)	Object Avoidance Distance (*m*)
Baseline	**No Protection**	46.5	164	5.00
Software	**Anomaly Detection**	24.2	245	5.47
	Temporal Redundancy	11.7	347	6.05
Hardware	**Modular Redundancy**	0	164	5.00
	Checkpointing	0	610	7.56
Adaptive Protection Paradigm (*VPP*) Front-end Software + Back-end Hardware		0	173	5.05



Takeaway: VPP reduce end-to-end compute latency overhead and reduce obstacle avoidance distance.

Evaluation – Autonomous Vehicle

Fault Protection Scheme		Resilience	Latency and Object Distance		Power Consumption and Driving Time	
		Error Propagation Rate (%)	Compute Latency (*ms*)	Object Avoidance Distance (*m*)	AD Component Power (*W*)*	AD Energy Change (%)
Baseline	**No Protection**	46.5	164	5.00	175	–
Software	**Anomaly Detection**	24.2	245	5.47	175	+33.14
	Temporal Redundancy	11.7	347	6.05	175	+75.24
Hardware	**Modular Redundancy**	0	164	5.00	473	+170.29
	Checkpointing	0	610	7.56	324	+91.52
Adaptive Protection Paradigm (*VPP*) Front-end Software + Back-end Hardware		0	173	5.05	175	+4.09

* The vehicle power without autonomous driving (AD) system is 600 W.

Takeaway: VPP reduce autonomous driving compute power and energy overhead.

Evaluation – Autonomous Vehicle

Fault Protection Scheme		Resilience	Latency and Object Distance		Power Consumption and Driving Time			
		Error Propagation Rate (%)	Compute Latency (*ms*)	Object Avoidance Distance (*m*)	AD Component Power (*W*)*	AD Energy Change (%)	Driving Time (hour)	Revenue Loss (%)
Baseline	**No Protection**	46.5	164	5.00	175	–	7.74	–
Software	**Anomaly Detection**	24.2	245	5.47	175	+33.14	7.20	-6.99
	Temporal Redundancy	11.7	347	6.05	175	+75.24	6.62	-14.52
Hardware	**Modular Redundancy**	0	164	5.00	473	+170.29	5.59	-27.78
	Checkpointing	0	610	7.56	324	+91.52	6.42	-17.13
Adaptive Protection Paradigm (*VPP*) Front-end Software + Back-end Hardware		0	173	5.05	175	+4.09	7.67	-0.92

* The vehicle power without autonomous driving (AD) system is 600 W.

Takeaway: VPP reduce autonomous driving compute power and energy overhead, thus enable longer driving time.

Evaluation – Autonomous Vehicle

Fault Protection Scheme		Resilience	Latency and Object Distance		Power Consumption and Driving Time				Cost
		Error Propagation Rate (%)	Compute Latency (*ms*)	Object Avoidance Distance (*m*)	AD Component Power (*W*)*	AD Energy Change (%)	Driving Time (hour)	Revenue Loss (%)	Extra Dollar Cost
Baseline	**No Protection**	46.5	164	5.00	175	–	7.74	–	–
Software	**Anomaly Detection**	24.2	245	5.47	175	+33.14	7.20	-6.99	negligible
	Temporal Redundancy	11.7	347	6.05	175	+75.24	6.62	-14.52	negligible
Hardware	**Modular Redundancy**	0	164	5.00	473	+170.29	5.59	-27.78	(CPU + GPU)×2
	Checkpointing	0	610	7.56	324	+91.52	6.42	-17.13	(CPU + GPU)×1
Adaptive Protection Paradigm (*VPP*) Front-end Software + Back-end Hardware		0	173	5.05	175	+4.09	7.67	-0.92	negligible

* The vehicle power without autonomous driving (AD) system is 600 W.

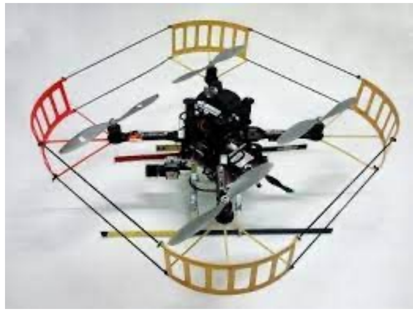
Takeaway: VPP reduces compute latency, energy and system overhead by taking advantage of (1) low cost and false-positive detection in front-end and (2) low latency in back-end. Conventional "one-size-fits-all" techniques are limited by tradeoffs in resilience and overhead.

Evaluation – Autonomous Drone

Fault Protection Scheme		Resilience	Latency and Flight Time			Power Consumption and Flight Energy				Cost
		Mission Failure Rate (%)	Compute Latency (ms)	Avg. Flight Velocity (m/s)	Mission Time (s)	Compute Power (W)	Mission Energy (kJ)	Num. of Missions	Endurance Reduction (%)	Extra Dollar Cost
Baseline	**No Protection**	12.20	871	2.79	107.53	15	60.09	5.62	–	–
Software	**Anomaly Detection**	6.44	1201	2.51	119.52	15	66.79	5.05	-10.04	negligible
	Temporal Redundancy	3.02	1924	2.14	140.18	15	78.34	4.31	-23.30	negligible
Hardware	**Modular Redundancy**	0	871	2.74	109.49	45	63.13	5.34	-3.79	TX2×2
	Checkpointing	0	3458	1.75	171.43	30	96.76	3.49	-37.90	TX2×1
Adaptive Protection Design Paradigm Frontend Software + Backend Hardware		0	897	2.77	108.30	15	60.52	5.58	-0.72	negligible

Experimental Setup
- Platform: Autonomous Drone (MAVBench[2])
- Reliability: soft errors

[2] Boroujerdian et al, MICRO, 2018



Evaluation – Autonomous Drone

Fault Protection Scheme		Resilience	Latency and Flight Time			Power Consumption and Flight Energy				Cost
		Mission Failure Rate (%)	Compute Latency (*ms*)	Avg. Flight Velocity (*m*/*s*)	Mission Time (*s*)	Compute Power (*W*)	Mission Energy (*kJ*)	Num. of Missions	Endurance Reduction (%)	Extra Dollar Cost
Baseline	**No Protection**	12.20	871	2.79	107.53	15	60.09	5.62	–	–
Software	**Anomaly Detection**	6.44	1201	2.51	119.52	15	66.79	5.05	-10.04	negligible
	Temporal Redundancy	3.02	1924	2.14	140.18	15	78.34	4.31	-23.30	negligible
Hardware	**Modular Redundancy**	0	871	2.74	109.49	45	63.13	5.34	-3.79	TX2×2
	Checkpointing	0	3458	1.75	171.43	30	96.76	3.49	-37.90	TX2×1
Adaptive Protection Design Paradigm Frontend Software + Backend Hardware		0	897	2.77	108.30	15	60.52	5.58	-0.72	negligible

Takeaway: For small form factor autonomous machines (e.g., drones), extra compute latency and payload weight brought by fault protection schemes impact drone safe flight velocity, further impacting end-to-end system mission time, mission energy, and flight endurance.

Evaluation – Autonomous Drone

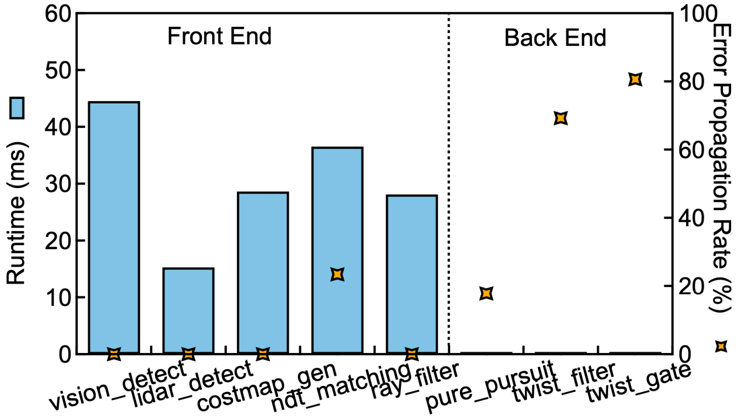
Fault Protection Scheme		Resilience	Latency and Flight Time			Power Consumption and Flight Energy				Cost
		Mission Failure Rate (%)	Compute Latency (ms)	Avg. Flight Velocity (m/s)	Mission Time (s)	Compute Power (W)	Mission Energy (kJ)	Num. of Missions	Endurance Reduction (%)	Extra Dollar Cost
Baseline	**No Protection**	12.20	871	2.79	107.53	15	60.09	5.62	–	–
Software	**Anomaly Detection**	6.44	1201	2.51	119.52	15	66.79	5.05	-10.04	negligible
	Temporal Redundancy	3.02	1924	2.14	140.18	15	78.34	4.31	-23.30	negligible
Hardware	**Modular Redundancy**	0	871	2.74	109.49	45	63.13	5.34	-3.79	TX2×2
	Checkpointing	0	3458	1.75	171.43	30	96.76	3.49	-37.90	TX2×1
Adaptive Protection Design Paradigm Frontend Software + Backend Hardware		0	897	2.77	108.30	15	60.52	5.58	-0.72	negligible

Takeaway: VPP generalizes well to small-scale drone system ***with improved resilience and negligible overhead***. By contrast, the large overhead from conventional "one-size-fits-all" protection results in severer performance degradation in SWaP-constrained systems.

Summary

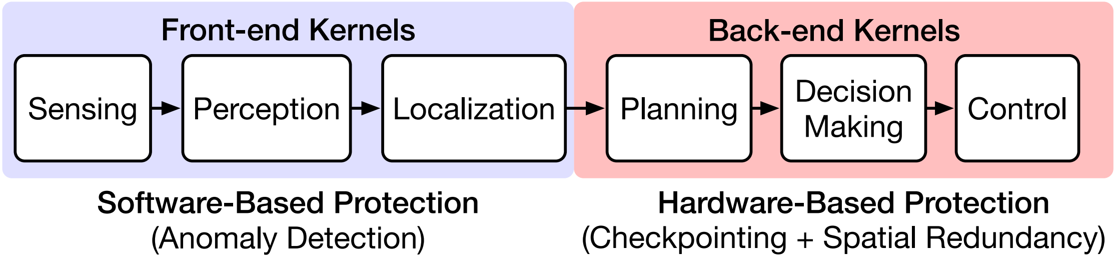
System Resiliency Characterization

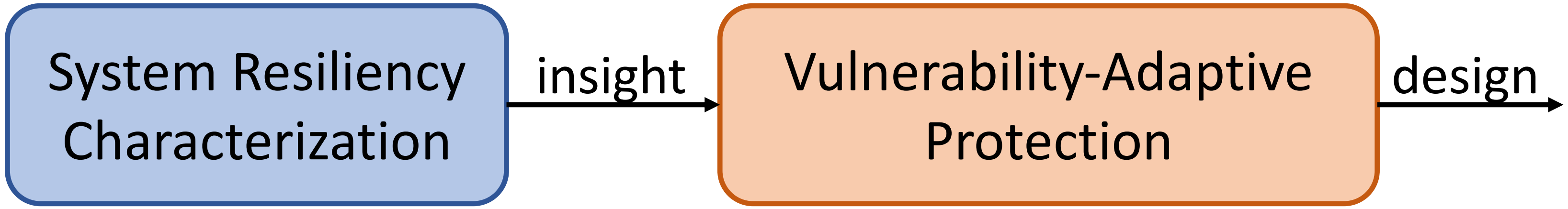
insight

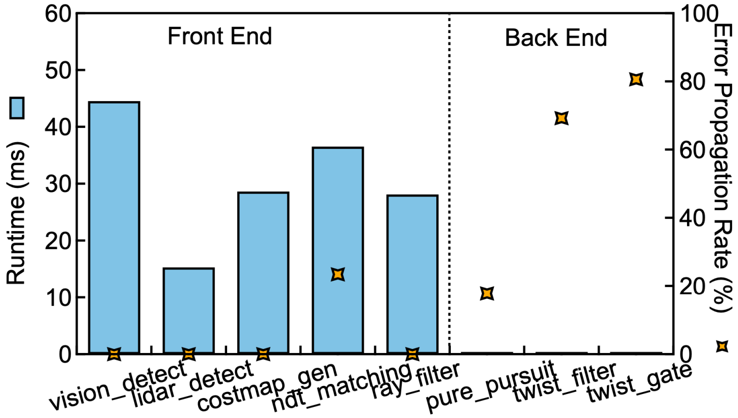


Inherent resiliency variations

Summary

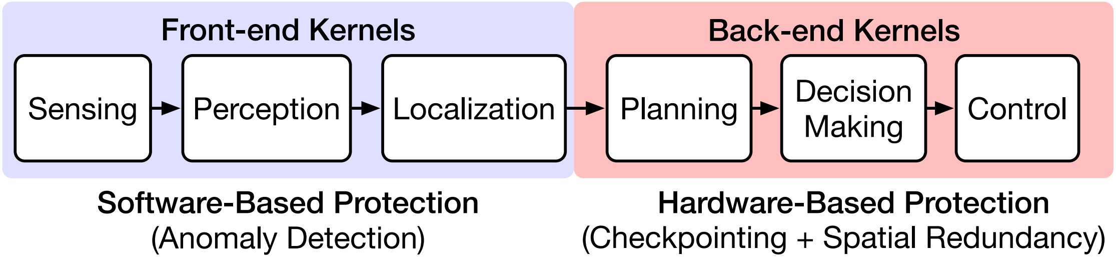


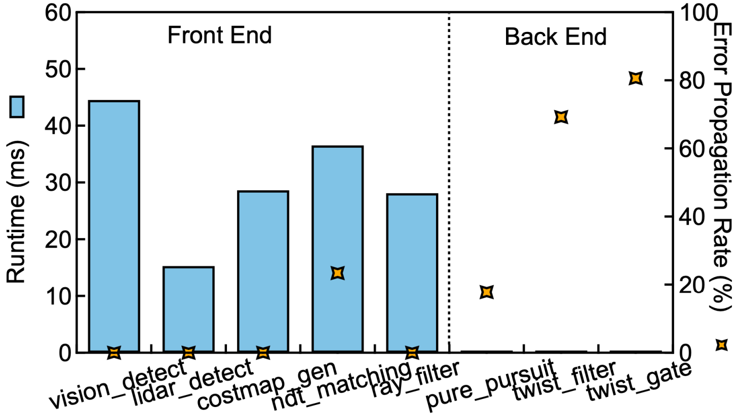




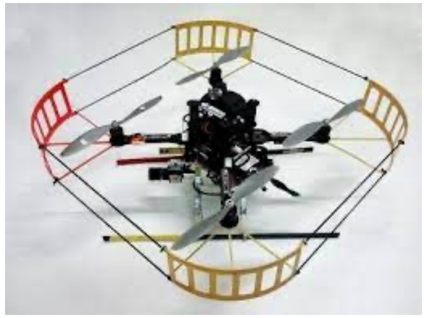
Inherent resiliency variations

Summary





Inherent resiliency variations



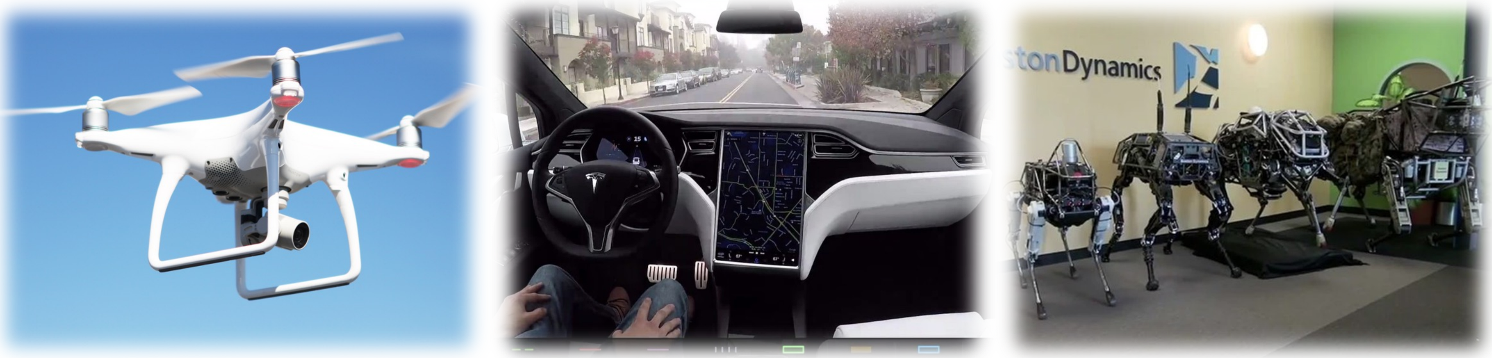
Resiliency improvement with low overhead

Summary

Goal: Improve operational resiliency under faults without degrading performance and efficiency (Vulnerable-Proportional Protection)

Reliability

Autonomous Machines



Performance

Performance-Efficiency-Reliability Co-Optimization

Efficiency

Goal: Improve task accuracy (Autonomy Algorithms)

Goal: Improve data and compute efficiency (Hardware Architecture)

COMMUNICATIONS OF THE ACM
CACM.ACM.ORG

Moving Towards Reliable Autonomous Machines: The Vulnerability-Adaptive Protection Paradigm

Zishen Wan[1*], Yiming Gan[2*], Bo Yu[3], Shaoshan Liu[3], Arijit Raychowdhury[1], Yuhao Zhu[2]

[1]Georgia Institute of Technology [2]University of Rochester
[3]Shenzhen Institute of AI and Robotics for Society
*(*Equal Contributions)*

zishenwan@gatech.edu, shaoshanliu@cuhk.edu.cn

Research and Advances, Communication of the ACM