

# Guaranteeing Pro-Active and Reactive Safety in intersections through resource management at the Edge

Matthias Becker

DIVERSE — 5th Workshop on Advanced Technologies in Industrial Vehicular Systems  
10. September 2024, Padua, Italy.

# Challenges for Urban Autonomous Driving

- A majority of AV crashes occur at intersections.
- The most considered influence on accidents in urban intersections is occlusion.
  - Parked vehicles
  - Stopped vehicles
  - Buildings
  - Construction
- Perception of AV alone is not sufficient in urban intersections, or decreases speed/performance



Hornsgatan, Stockholm (Source: Google Maps)

# Overview of the Talk

## Background

## Towards increased safety at smart intersections

- Augmented AV perception
- Pro-active safety through traffic orchestration
- Reactive safety through contingency path planning

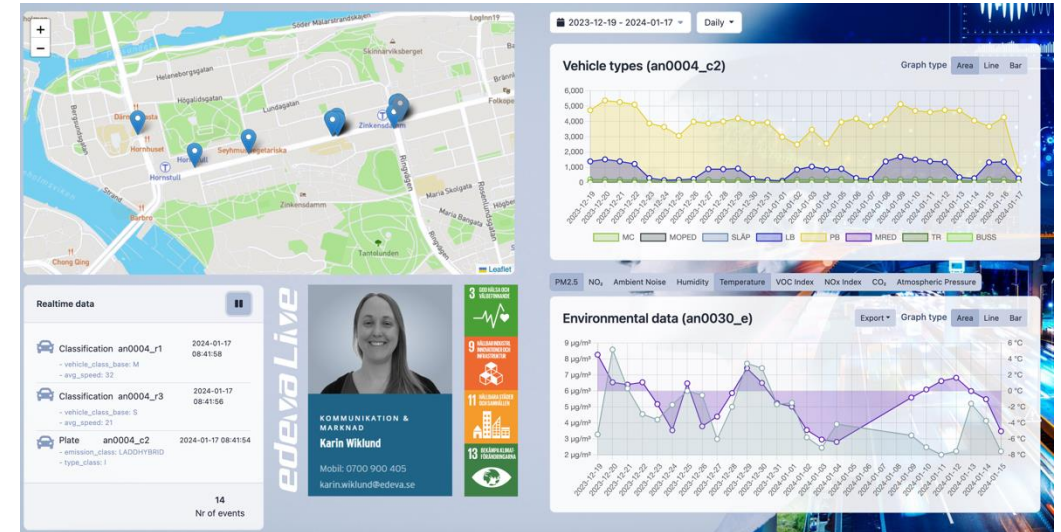
## Platform support to meet workload demands

- Scheduling reservations to isolate applications
- Request arbitration under different types of timing constraints

## Conclusions and future work

# Holistic Information at Intersections

- In addition to cameras for perception, various data can be available at intersections
  - Vehicle Class
  - Emission Class
  - Speed
  - Weight
  - etc.
- Active measures exist to influence traffic dynamically



Source: <https://live.edeva.se/open-dashboard>

# Infrastructure Guided Decision Making Support

Roadside Unit (RSU) provides resources to AVs:

- Augmented AV perception
- Traffic orchestration to avoid congestion at intersection
- Contingency path planning during likely collisions and establish safety zones

Proactive

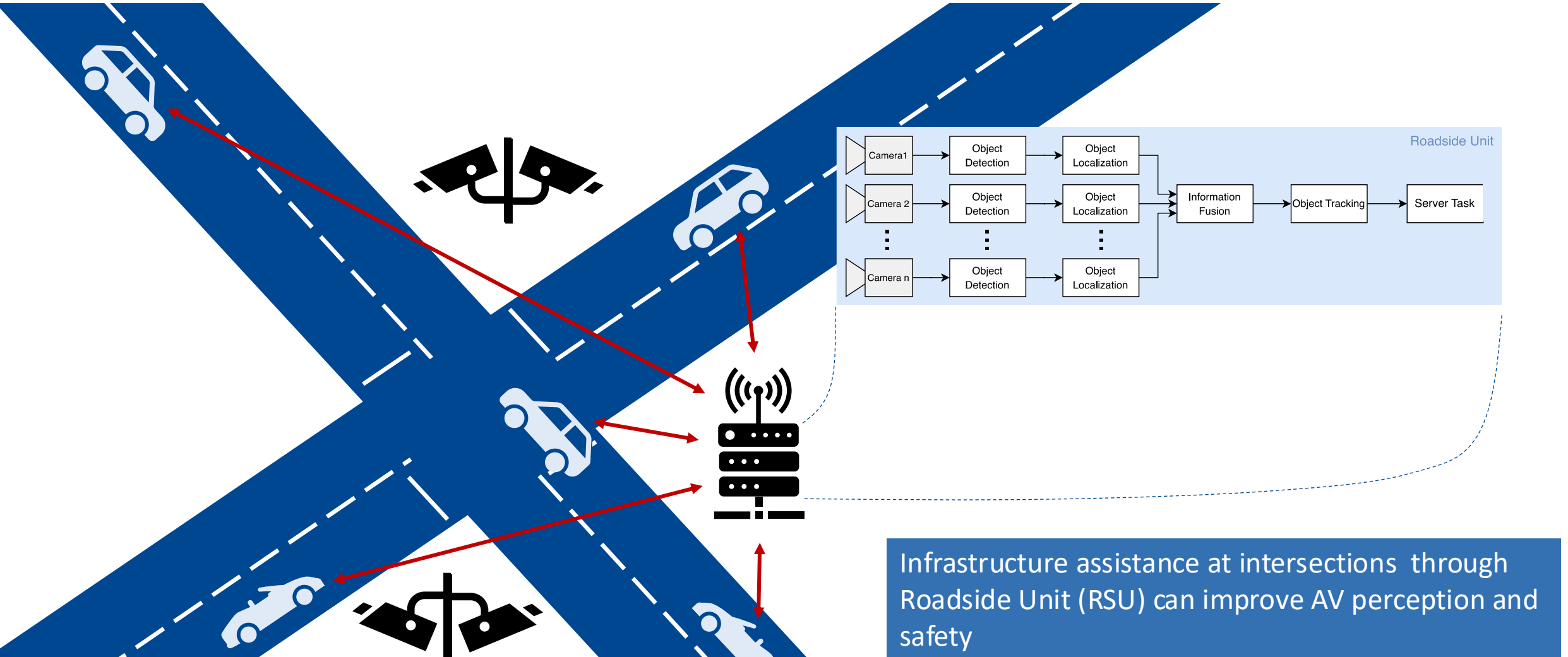
Reactive

- Path Planning
- Establish Safety Zones
- Scene Perception

Safe Infrastructure Guided  
Decision Making Support

- Computational resources for different workloads
- Handle AV requests with timing constraints

# Improve AV Perception through Infrastructure Support



# Traffic Orchestration

## Time-to-Collision (TTC)

“The time that remains until a crash between two vehicles would have occurred if the crash course and speed difference are maintained.” Hayward, 1972.

Quantify the severity of a conflict as well as its probability.

→ Take evasive action if a threshold is exceeded.

## Utilise SSM for Traffic Orchestration

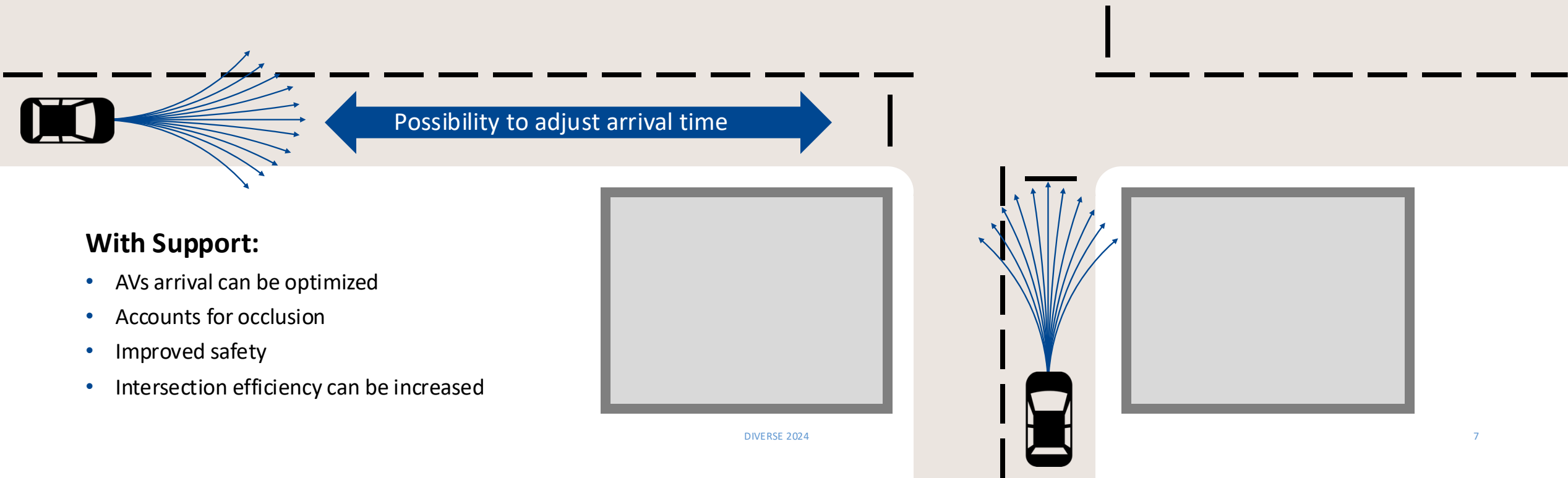
Orchestrate / support AV planning to arrive at intersection when likelihood of collision is minimal.

→ Reduce the likelihood of encounters that lead to difficult-to-control situations.



# Traffic Orchestration

- Provide advancing AVs with decision-making support to reduce the likelihood of conflicts at the intersection.
- Consider possible trajectories and their likelihood
- Recommend actions to AV that reduce likelihood of conflict



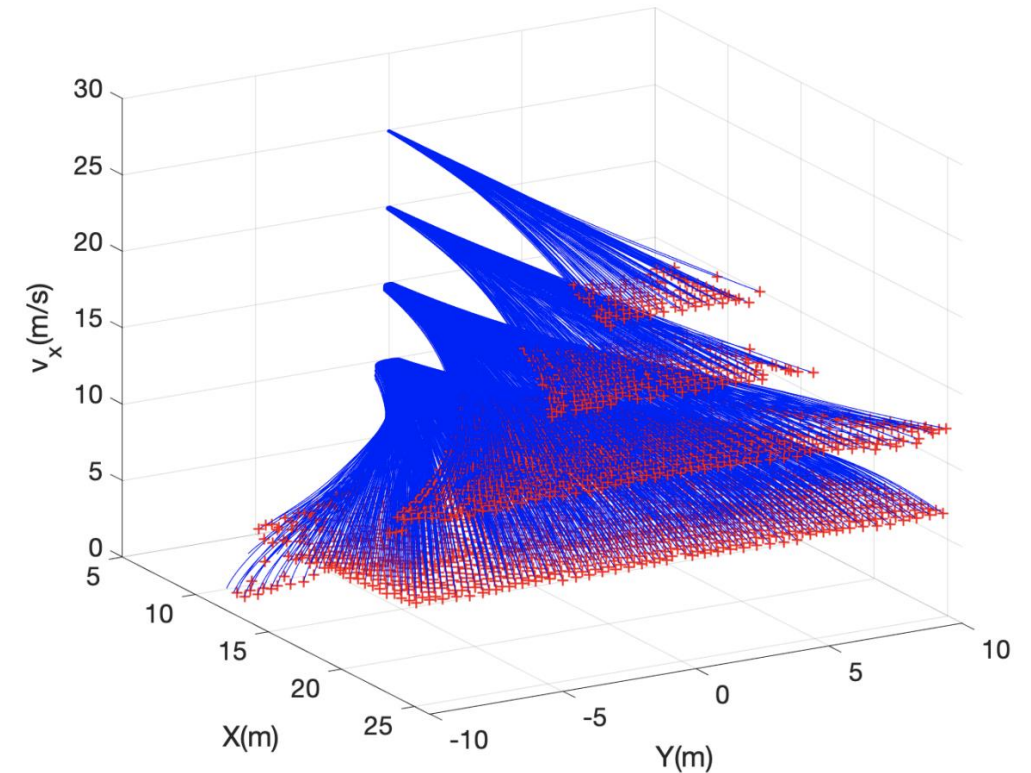


# Support Decision Making in Unavoidable Collisions

- Trajectory library is used to present possible evasive actions of the ego vehicle.
- Parameterized:
  - Initial velocity
  - Weight
  - Size
  - etc.

**Problem – Vehicle parameters are typically not known!**

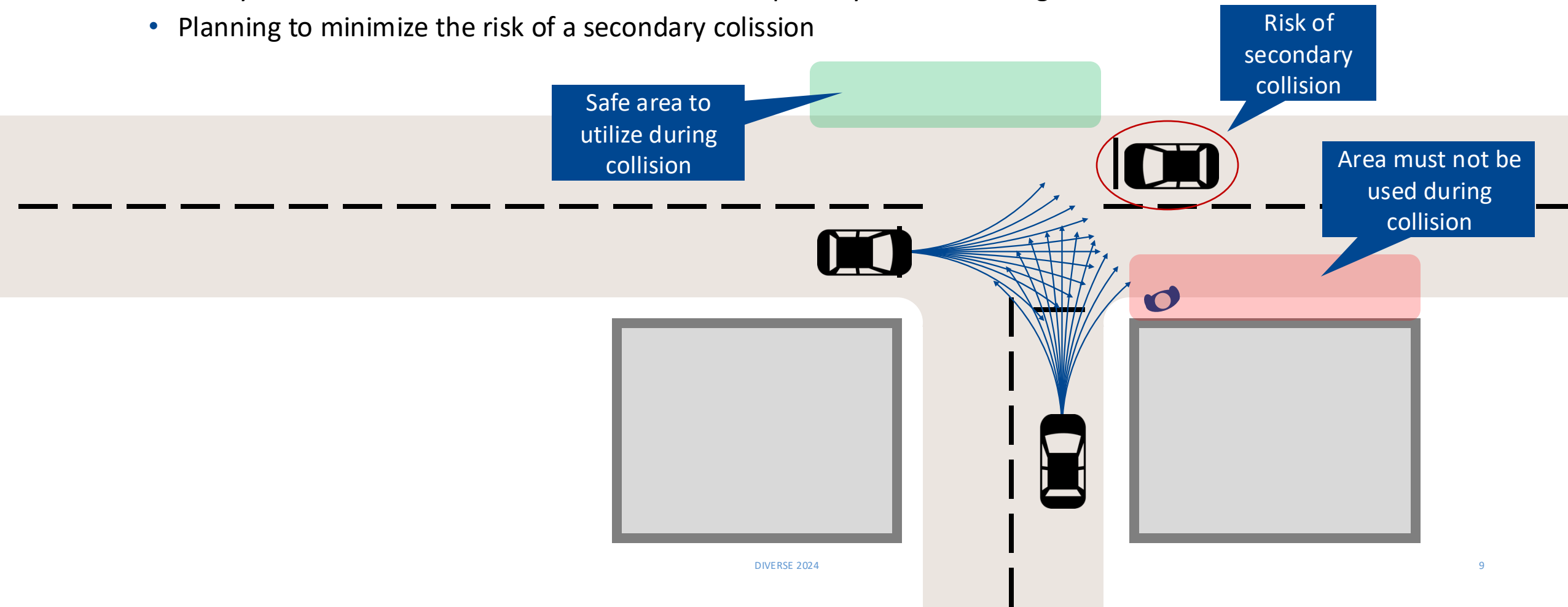
- → RSU at the intersection has more accurate data on vehicle parameter, involved actors, etc. than individual AV.



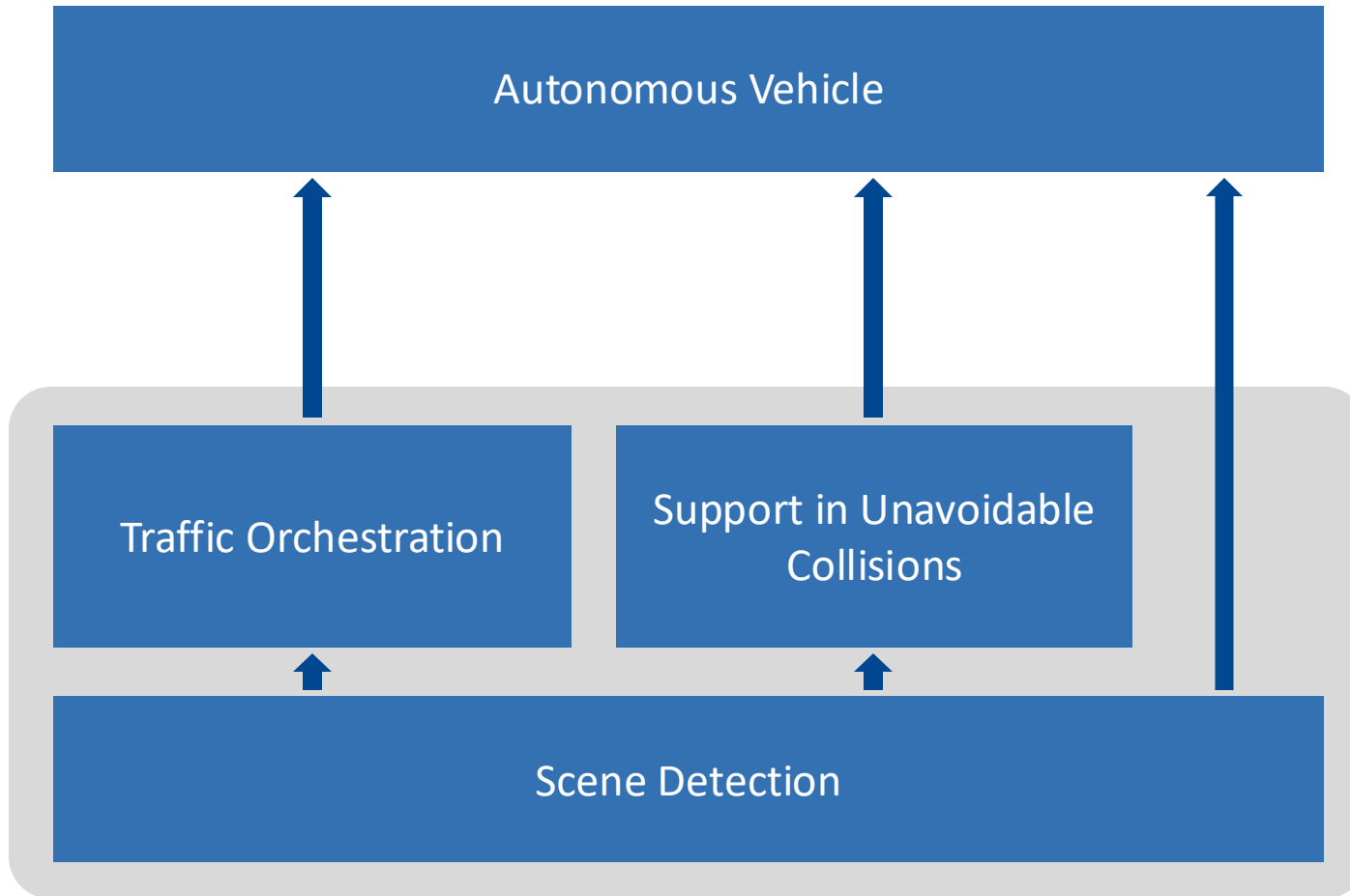
Source: Parseh et al., IEEE T-IV, 2021

# Support Decision Making in Unavoidable Collisions

- SSM indicates that the likelihood of collision is above threshold
- Intersection support can include:
  - Safety Zones, areas outside the road that can temporarily be used during crash
  - Planning to minimize the risk of a secondary collision



# Service Types on RSU



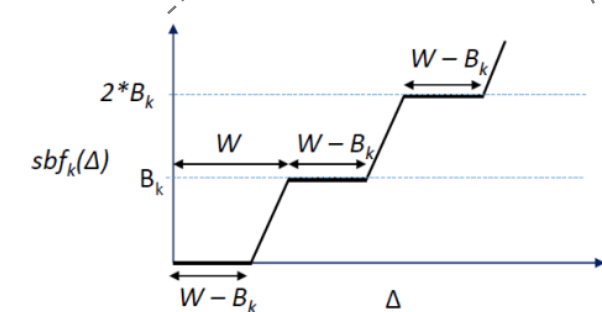
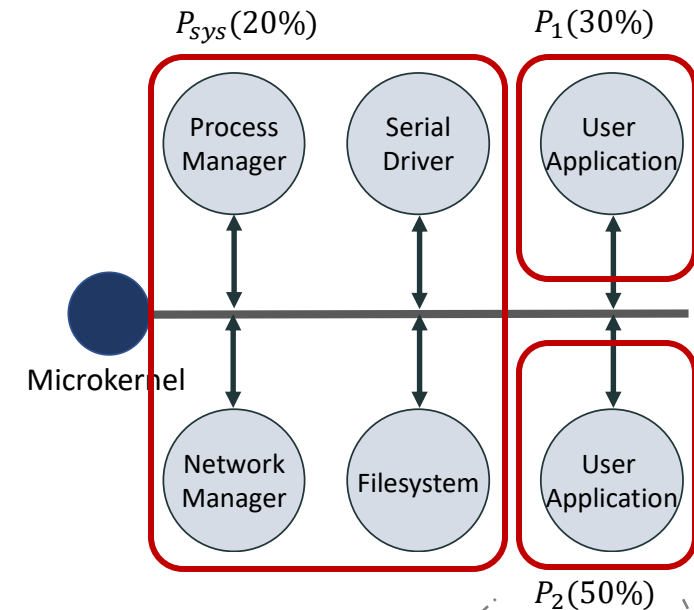
- Different services are provided by the RSU
- Runtime interference between services must be avoided
- Computationally heavy data-driven workload
- Different AVs can have different (types of) timing requirements

# POSIX-Based RTOS

- Workload Characteristics
  - Data-driven, dynamic workloads
  - Modular updates
- QNX RTOS
  - Preferred base operating system by many automotive OEMs
  - ISO26262 pre-certified at the highest level of assurance (ASIL-D)
  - POSIX-compliant, commercially proven
  - Supports CPU reservations

## Adaptive Partitioning Scheduler (APS)

- Secure partitions with guaranteed CPU time
- A partition is a container for a collection of threads
- Configured with budget as percentage of CPU time over a common sliding window (100 ms)
- The highest priority ready threads whose partition has budget is scheduled
- Budget reclamation can be enabled



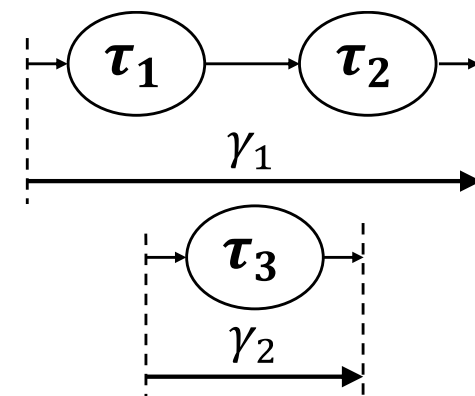
# Analysis vs. Measurements

Task	WCET	Period	Priority	Partition
$\tau_1$	20 ms	100 ms	253	$P_1$
$\tau_2$	10 ms	-	254	$P_1$
$\tau_3$	x ms	10 ms	255	$P_2$

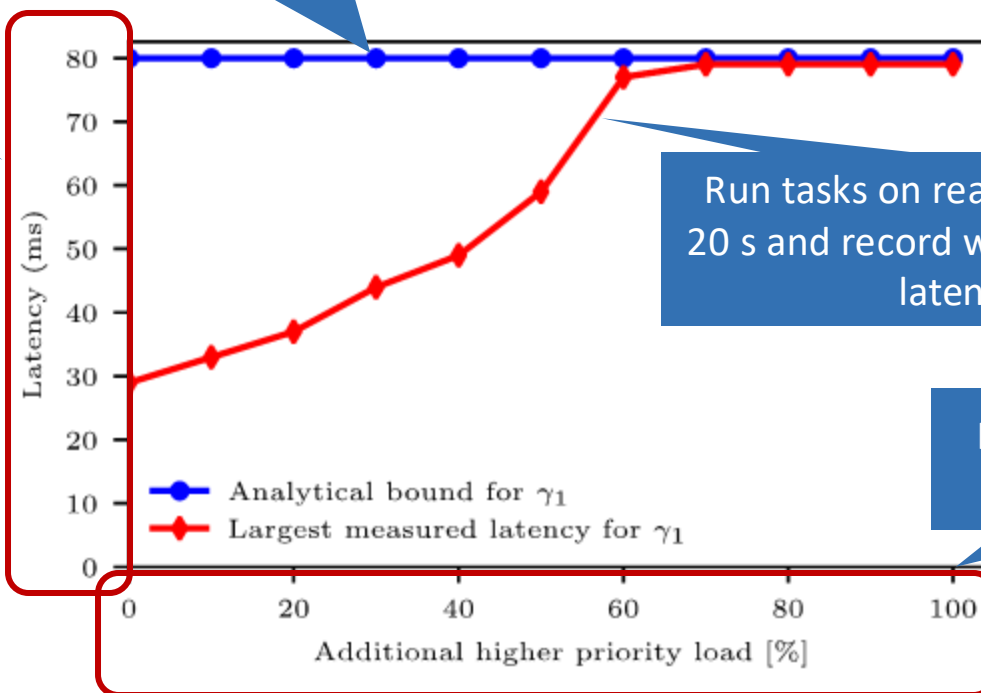
- 3 tasks, 2 event-driven chains
- All tasks on the same core
- Budget  $P_1$  and  $P_2$  is 50%
- budget reclaiming

Compute the latency bound using the analysis

Vary WCET to affect the task utilization



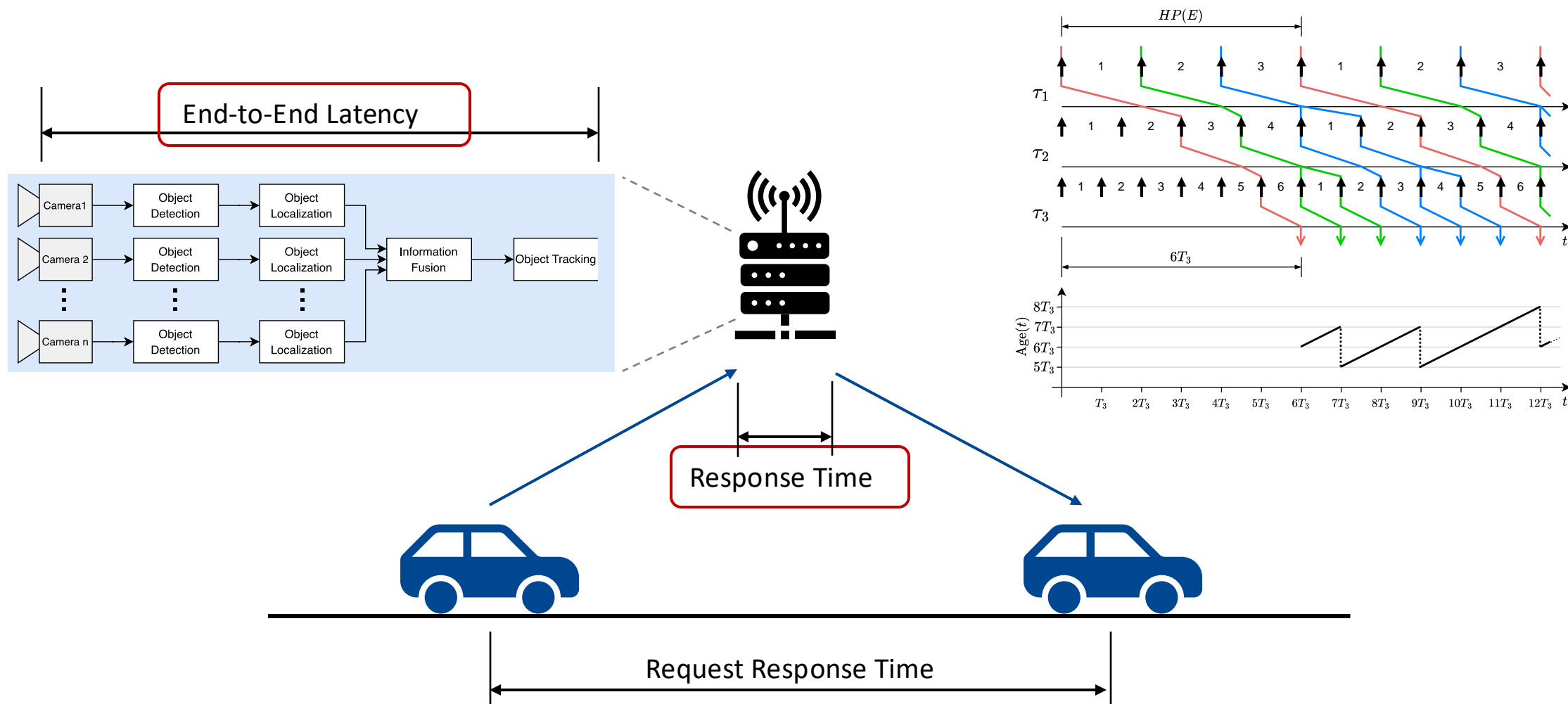
Latency of chain  $\gamma_1$



Run tasks on real platform for 20 s and record worst-observed latency

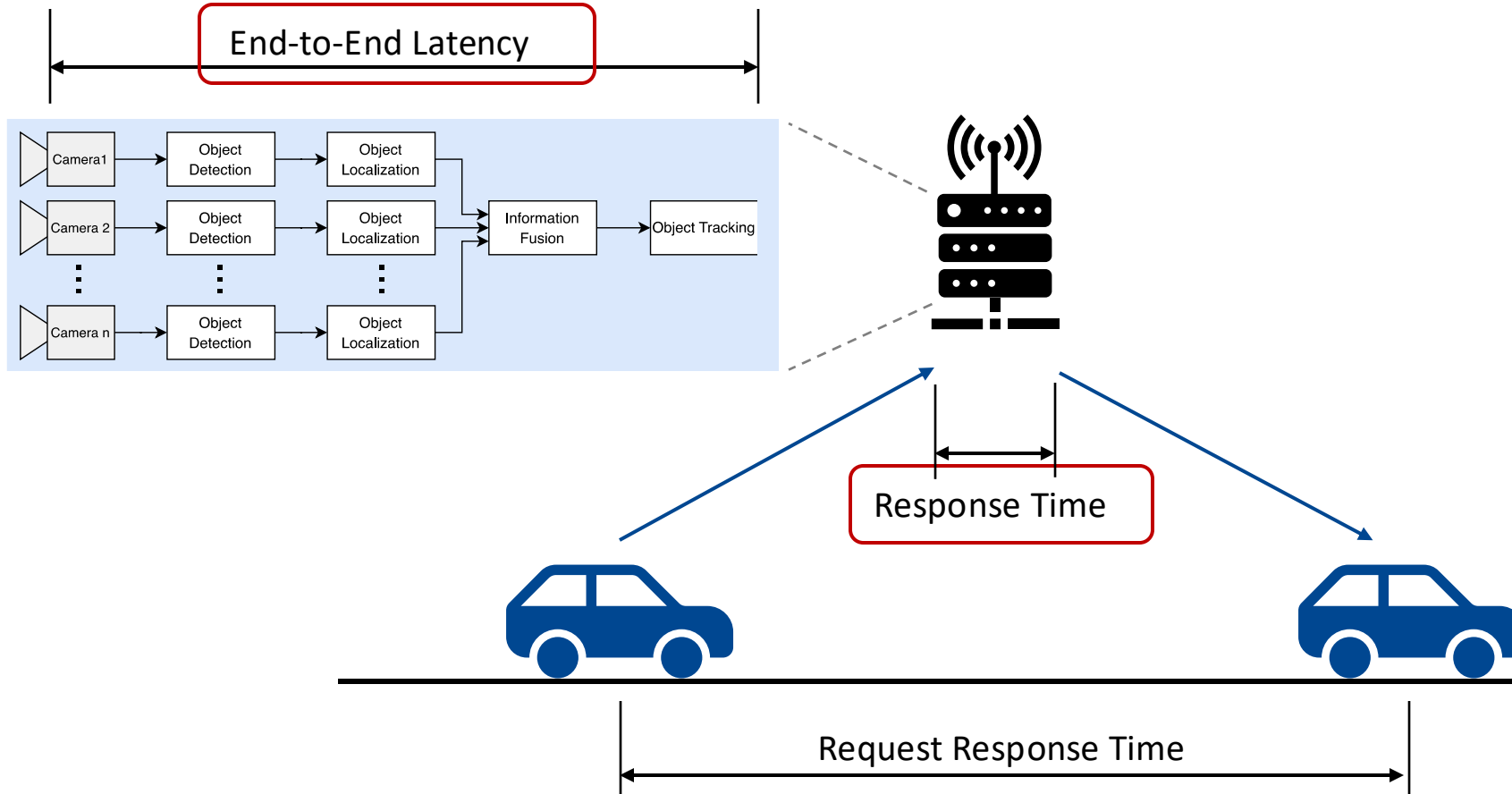
Higher priority load generated by  $\tau_3$

# Augmented AV perception

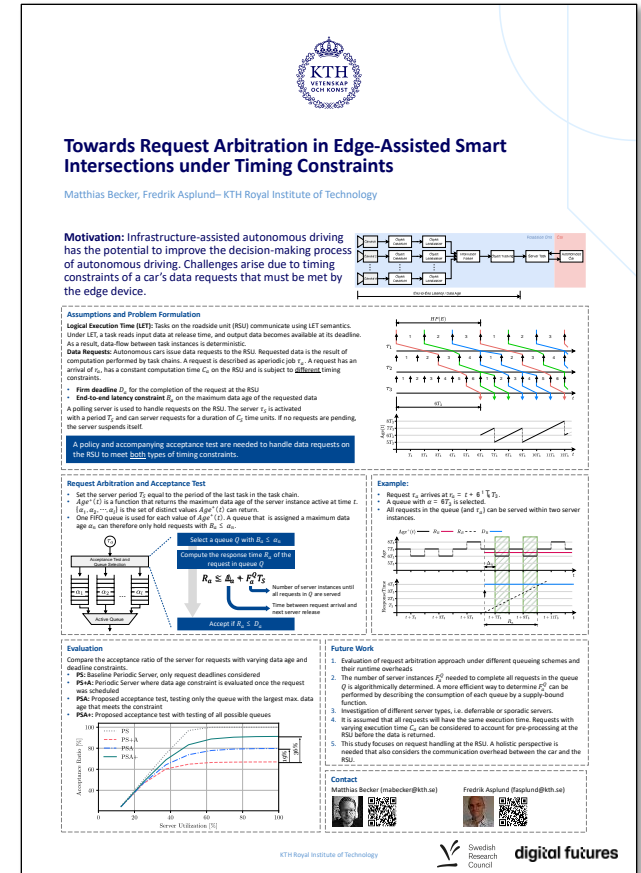


How can both types of timing constraints be considered by the RSU?

# Augmented AV perception



How can both types of timing constraints be considered by the RSU?



**Towards Request Arbitration in Edge-Assisted Smart Intersections under Timing Constraints**  
 Matthias Becker, Fredrik Asplund - KTH Royal Institute of Technology

**Motivation:** Infrastructure-assisted autonomous driving has the potential to improve the decision-making process of autonomous driving. Challenges arise due to timing constraints of a car's data requests that must be met by the edge device.

**Assumptions and Problem Formulation**  
 Logical Execution Time (LET): Tasks on the roadside unit (RSU) communicate using LET semantics. Under LET, a task reads input data at release time, and output data becomes available at its deadline. As a result, data flow between task instances is deterministic.  
 Data Requests: Autonomous cars issue data requests to the RSU. Requested data is the result of computation performed by task chains. A request is described as aperiodic job  $r_i$ . A request has an arrival of  $r_i$ , has a constant computation time  $C_i$  on the RSU and is subject to **strict** timing constraints.

- Flow deadline**  $D_i$  for the completion of the request at the RSU
- End-to-end latency constraint**  $E_i$  on the maximum data age of the requested data

A polling server is used to handle requests on the RSU. The server  $\tau_i$  is activated with a period  $T_i$  and can server requests for a duration of  $C_i$  time units. If no requests are pending, the server suspends itself.

A policy and accompanying acceptance test are needed to handle data requests on the RSU to meet **both** types of timing constraints.

**Request Arbitration and Acceptance Test**

- Set the server period  $T_i$  equal to the period of the last task in the task chain.
- $Age^i(t)$  is a function that returns the maximum data age of the server instance active at time  $t$ .  $\{a_1, a_2, \dots, a_n\}$  is the set of distinct values  $Age^i(t)$  can return.
- One FIFO queue is used for each value of  $Age^i(t)$ . A queue that is assigned a maximum data age  $a_i$  can therefore only hold requests with  $B_i \leq a_i$ .

Flowchart: Select a queue  $Q$  with  $B_i \leq a_i$  → Compute the response time  $R_i$  of the request in queue  $Q$  →  $R_i \leq S_i + R_i^q T_i$  → Accept if  $B_i \leq D_i$

**Evaluation**  
 Compare the acceptance ratio of the server for requests with varying data age and deadline constraints.  
 PSk: Periodic Server, only request deadline considered  
 PSk: Periodic Server where data age constraint is evaluated once the request was scheduled  
 PSA: Proposed acceptance test, testing only the queue with the largest max. data age that meets the constraint  
 PSk: Proposed acceptance test with testing of all possible queues

**Future Work**  
 1. Evaluation of request arbitration approach under different queuing schemes and their runtime overhead  
 2. The number of server instances  $P^i$  needed to complete all requests in the queue  $Q$  is algorithmically determined. A more efficient way to determine  $P^i$  can be performed by describing the consumption of each queue by a supply-bound function.  
 3. Investigation of different server types, i.e. deferrable or sporadic servers.  
 4. It is assumed that all requests will have the same execution time. Requests with varying execution time  $C_i$  can be considered to account for pre-processing at the RSU before the data is returned.  
 5. This study focuses on request handling at the RSU. A holistic perspective is needed that also considers the communication overhead between the car and the RSU.

**Contact**  
 Matthias Becker (mbecker@kth.se) | Fredrik Asplund (faspund@kth.se)

KTH Royal Institute of Technology | Swedish Research Council | digital futures

→ Thursday 9:00 WIP3



# Conclusions and Future Work

- Infrastructure-assisted autonomous driving has the potential to reduce risks at intersections.
- Holistic view of the traffic allows for pro-active and reactive approaches for safety
- Different services for AVs emerge that are subject to timing constraints
- CPU reservations to isolate different applications/services from each other
- Server-based AV request handling considering different types of timing constraints
- Simulation study to:
  - Evaluate effectiveness of Remote-SSM
  - Identify realistic timing constraints for AV traffic at intersection

# Thank You! Questions?