CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Active Monitoring with RTLola

A Specification-Guided Scheduling Approach

Jan Baumeister, Bernd Finkbeiner, ***Frederik Scheerer***

RV 2025 | September 18, 2025

Motivation

Photo: DLR (CC BY-NC-ND 3.0)

Motivation

GPS Sensor

Altitude Sensor

Camera

Photo: DLR (CC BY-NC-ND 3.0)

Monitor

Motivation

Photo: DLR (CC BY-NC-ND 3.0)

Monitoring Drone Sensors: Low Frequency

Video

Motivation

Photo: DLR (CC BY-NC-ND 3.0)

Limited Bandwidth

because restrictions concerning energy consumption, size or weight.

Motivation

Photo: DLR (CC BY-NC-ND 3.0)

Motivation

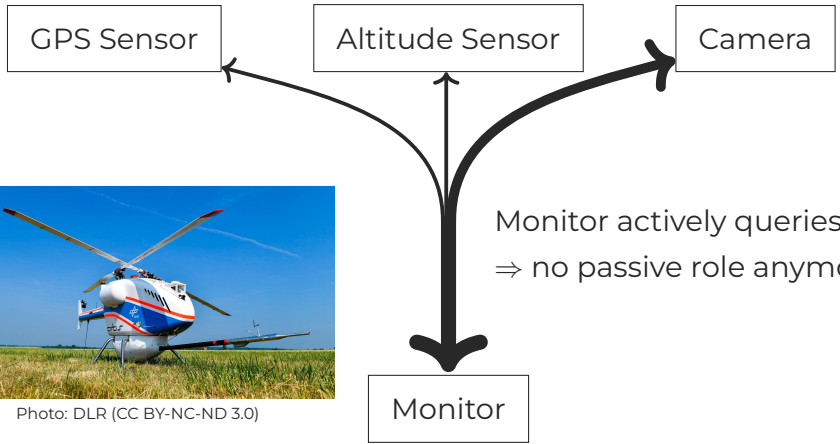


Photo: DLR (CC BY-NC-ND 3.0)

Monitoring Drone Sensors: Scheduled

Video

Monitor Semantics

Evaluation Model $\mathbb{W}$ is valid, if:

- Values are calculated according to specification
- Timestamps are monotonically increasing

Scheduled Monitor Semantics

Evaluation Model $\mathbb{W}$ is valid, if:

- Values are calculated according to specification
- Timestamps are monotonically increasing, and
- Inputs arrive according to **scheduling constraint**.

Dynamic Scheduling Constraints

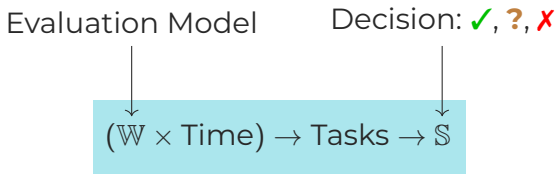
$$(\mathbb{W} \times \text{Time}) \to \text{Tasks} \to \mathbb{S}$$

Dynamic Scheduling Constraints

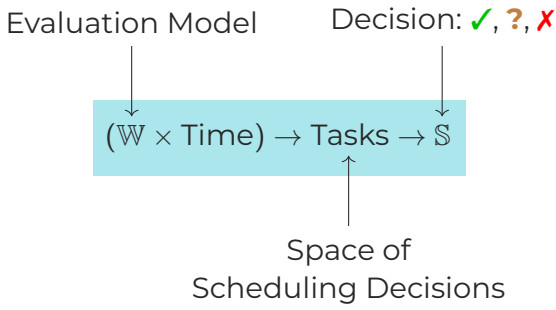
Evaluation Model

$$(\mathbb{W} \times \text{Time}) \to \text{Tasks} \to \mathbb{S}$$

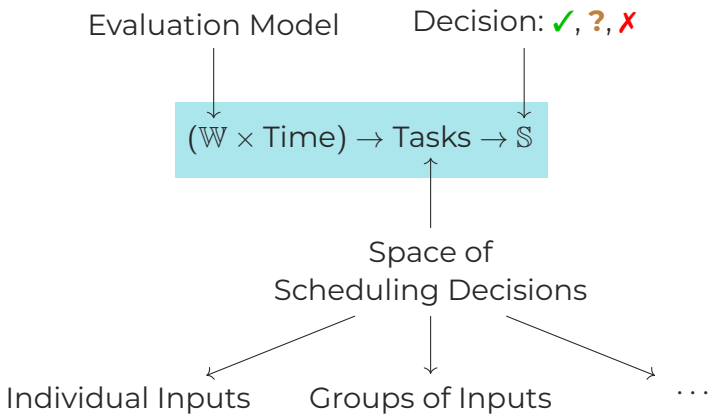
Dynamic Scheduling Constraints



Dynamic Scheduling Constraints



Dynamic Scheduling Constraints



Scheduling Constraints

Dynamic Constraint

$$(\mathbb{W} \times \text{Time}) \rightarrow \text{Tasks} \rightarrow \mathbb{S}$$

Requires access to the whole evaluation model

Scheduling Constraints

Dynamic Constraint

$$(\mathbb{W} \times \text{Time}) \to \text{Tasks} \to \mathbb{S}$$

Requires access to the whole evaluation model

Static Constraint

$$\text{Tasks} \to \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Can be evaluated during runtime

Scheduling Constraints

Dynamic Constraint

$$(\mathbb{W} \times \text{Time}) \to \text{Tasks} \to \mathbb{S}$$

Requires access to the whole evaluation model

Static Constraint

$$\text{Tasks} \to \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Can be evaluated during runtime

Translated by the scheduling algorithm

Static Constraints

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Static Constraints

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Deadline

$\mathbb{V}$ are time durations:
e.g. 100ms, 1s, 5s

maximum duration
between evaluations

Static Constraints

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Deadline	**Priority**
$\mathbb{V}$ are time durations: e.g. 100ms, 1s, 5s	$\mathbb{V}$ are priorities: high, medium, low
maximum duration between evaluations	highest priority tasks are evaluated first

Static Constraints

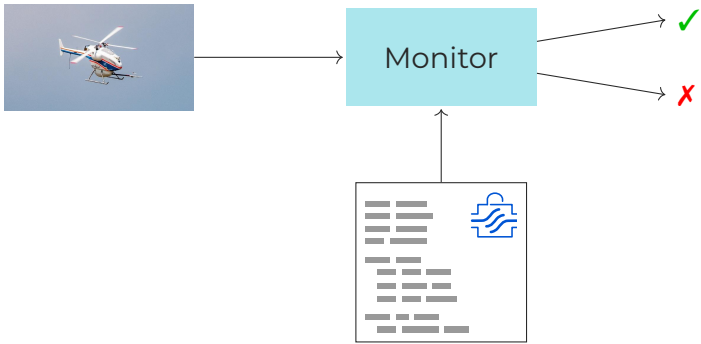
$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Deadline	**Priority**	**Priority + Deadline**
$\mathbb{V}$ are time durations: e.g. 100ms, 1s, 5s	$\mathbb{V}$ are priorities: high, medium, low	$\mathbb{V}$ are priorities: high, medium, low
maximum duration between evaluations	highest priority tasks are evaluated first	tasks become overdue after deadline

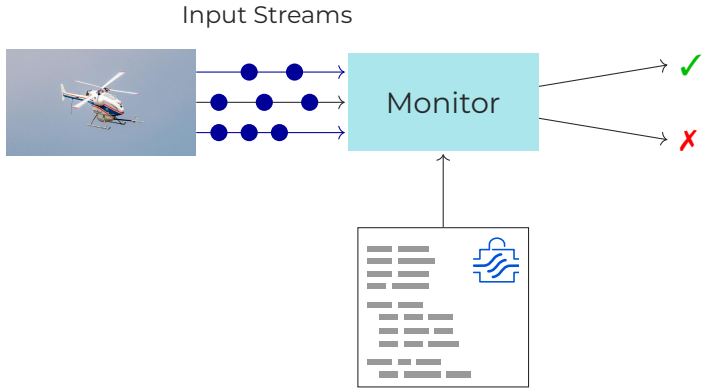
Monitor

✓

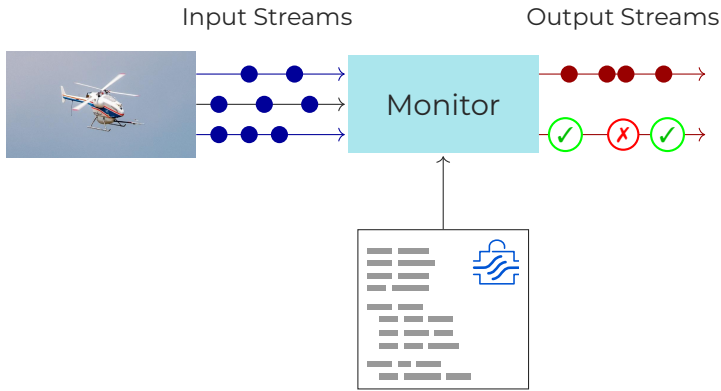
✗

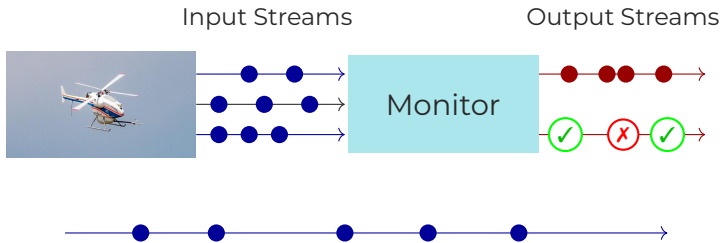

Monitor
✓
✗

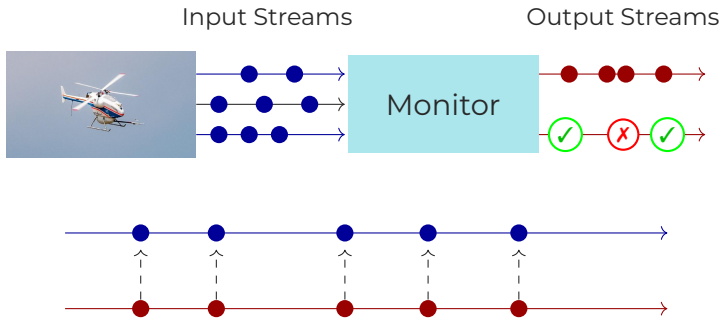
RTLola

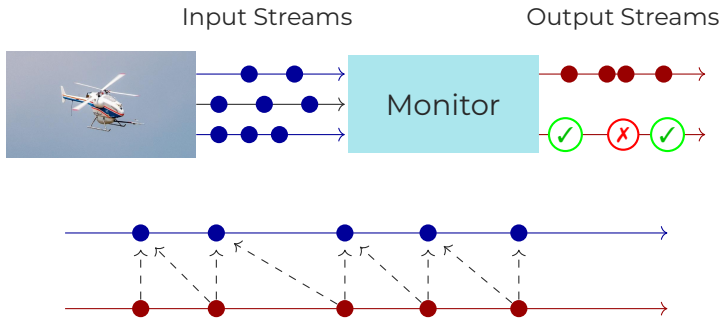


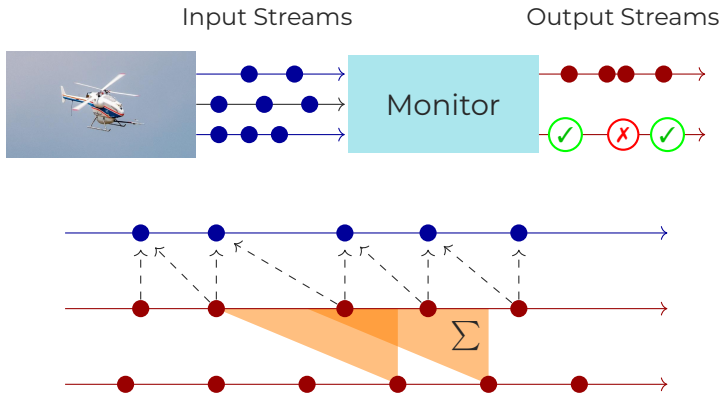
RTLola




Input Streams
Output Streams
Monitor

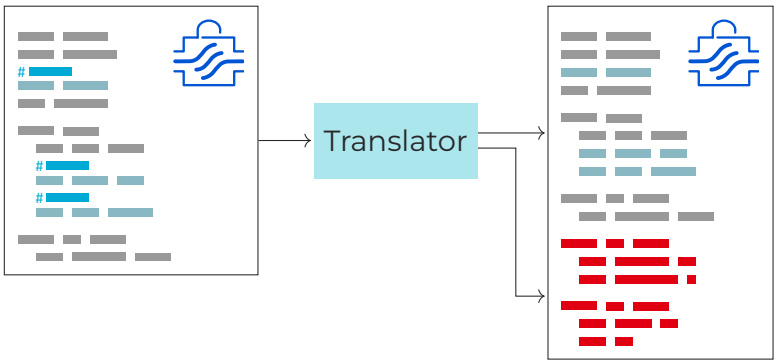

Input Streams
Output Streams
Monitor


Input Streams
Output Streams
Monitor

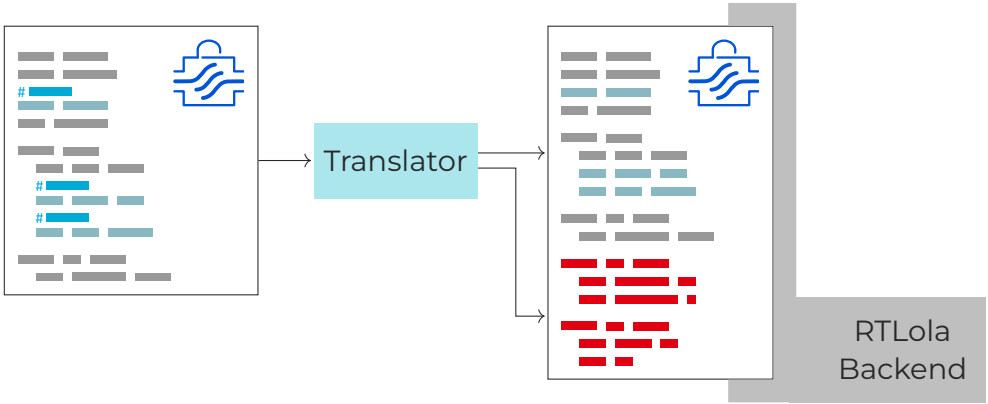

Input Streams
Output Streams
Monitor
✓
✗
✓
Σ

Overview

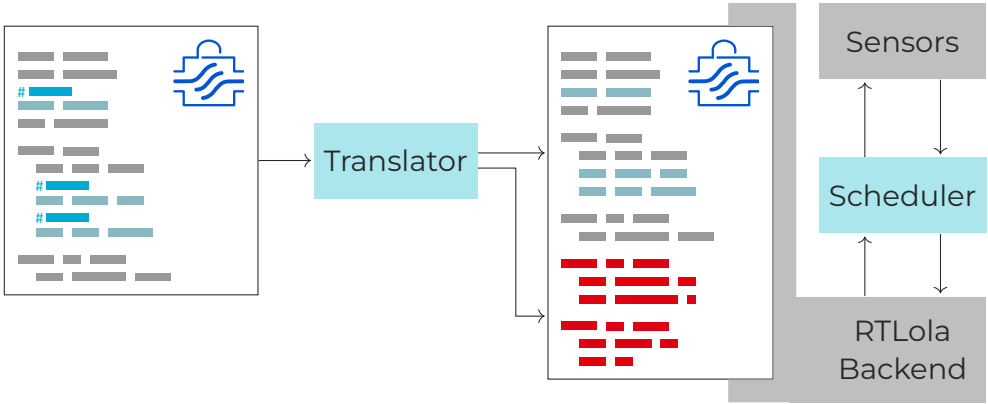
Overview



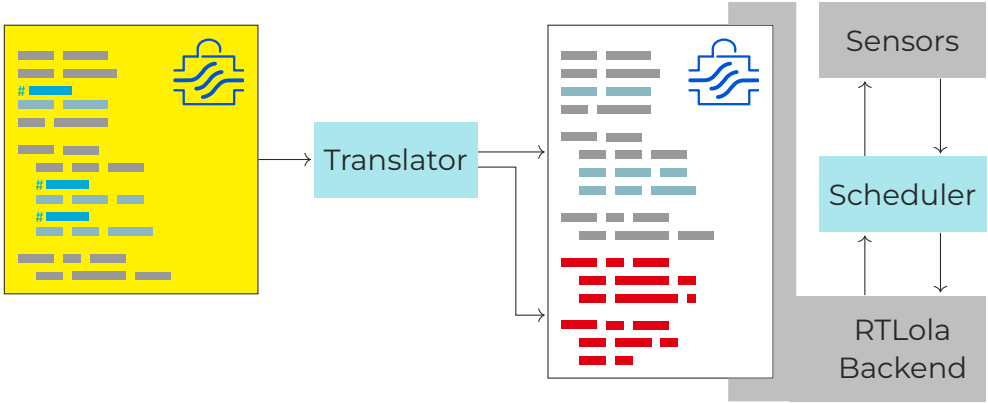
Overview



Overview



Overview



Scheduling Annotations

```
output dist @lat∧lon := ...
output dist_violation
  #[priority=low]
  eval @lat∧lon when dist < 20.0
                with false
  #[priority=medium]
  eval @lat∧lon when dist < 50.0
                with false
  #[priority=high]
  eval @lat∧lon when true
                with dist > 70.0
```

Scheduling Annotations

```
output dist @lat∧lon := ...
output dist_violation
  #[priority=low]
  eval @lat∧lon when dist < 20.0
                with false
  #[priority=medium]
  eval @lat∧lon when dist < 50.0
                with false
  #[priority=high]
  eval @lat∧lon when true
                with dist > 70.0
```

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Scheduling Annotations

```
output dist @lat∧lon := ...
output dist_violation
  #[priority=low]
  eval @lat∧lon when dist < 20.0
                with false
  #[priority=medium]
  eval @lat∧lon when dist < 50.0
                with false
  #[priority=high]
  eval @lat∧lon when true
                with dist > 70.0
```

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

@lat∧lon	
dist < 20	low

Scheduling Annotations

```
output dist @lat∧lon := ...
output dist_violation
  #[priority=low]
  eval @lat∧lon when dist < 20.0
                with false
  #[priority=medium]
  eval @lat∧lon when dist < 50.0
                with false
  #[priority=high]
  eval @lat∧lon when true
                with dist > 70.0
```

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

@lat∧lon	
dist < 20	low
20 ≤ dist < 50	medium

Scheduling Annotations

```
output dist @lat∧lon := ...
output dist_violation
  #[priority=low]
  eval @lat∧lon when dist < 20.0
                with false
  #[priority=medium]
  eval @lat∧lon when dist < 50.0
                with false
  #[priority=high]
  eval @lat∧lon when true
                with dist > 70.0
```

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

@lat∧lon	
dist < 20	low
20 ≤ dist < 50	medium
dist ≥ 50	high

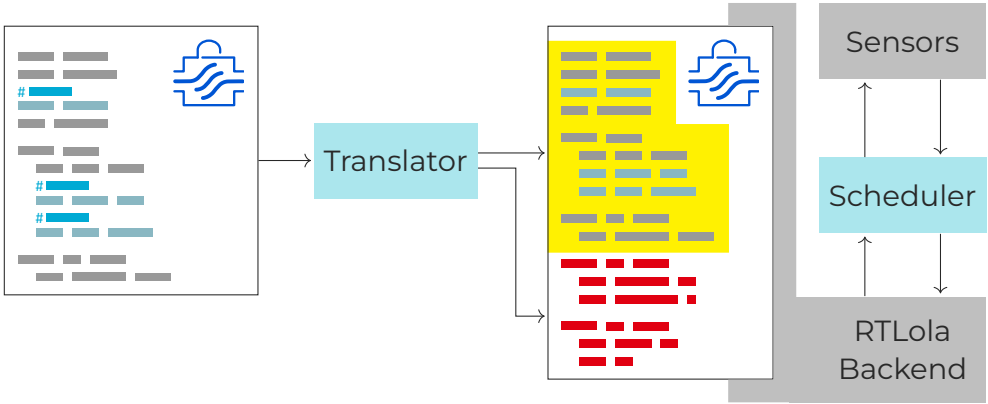
Scheduling Annotations

```
output dist @lat∧lon := ...
output dist_violation

  eval @lat∧lon when dist < 20.0
               with false

  eval @lat∧lon when dist < 50.0
               with false

  eval @lat∧lon when true
               with dist > 70.0
```

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

@lat∧lon	
dist < 20	low
20 ≤ dist < 50	medium
dist ≥ 50	high

Overview



Additional Streams

@lat∧lon	
dist < 20	low
20 ≤ dist < 50	medium
dist ≥ 50	high

Additional Streams

separate stream
for each task

@lat∧lon	
dist < 20	low
20 ≤ dist < 50	medium
dist ≥ 50	high

```
output schedule_lat∧lon
  eval @lat∧lon when dist ≥ 50 with high
  eval @lat∧lon when 20 ≤ dist < 50 with medium
  eval @lat∧lon when dist < 20 with low
```

Additional Streams

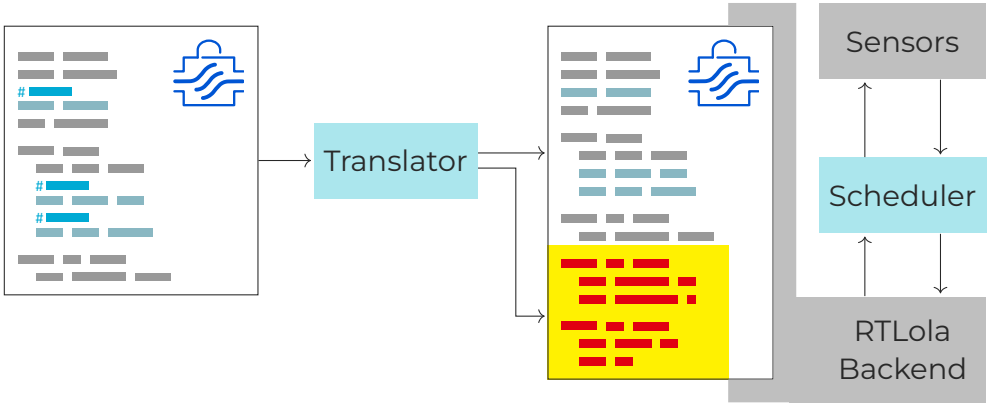
separate stream
for each task

@lat∧lon	
dist < 20	low
20 ≤ dist < 50	medium
dist ≥ 50	high

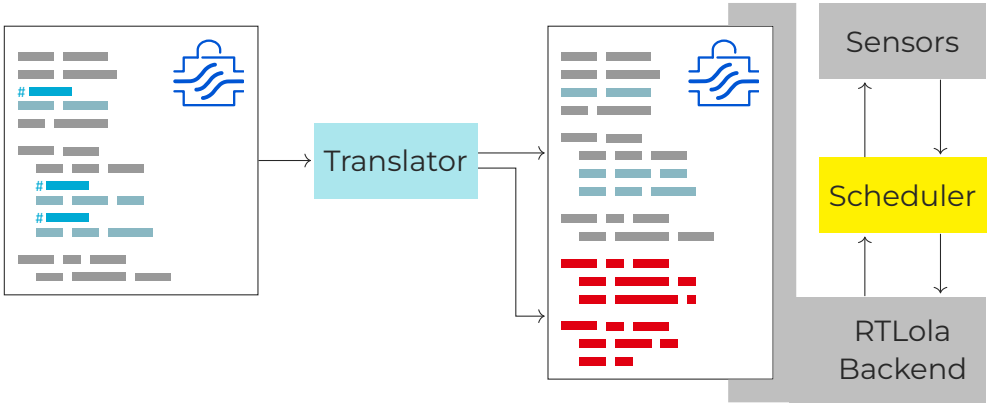
```
output schedule_lat∧lon
  eval @lat∧lon when dist ≥ 50 with high
  eval @lat∧lon when 20 ≤ dist < 50 with medium
  eval @lat∧lon when dist < 20 with low
```

sorted descendingly

Overview



Overview



Scheduling Algorithms

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Deadline **Priority** **Priority + Deadline**

Scheduling Algorithms

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Deadline

Fill available bandwidth with tasks that are due the soonest (Earliest Deadline First)

Priority

Priority + Deadline

Scheduling Algorithms

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Deadline	Priority	Priority + Deadline
Fill available bandwidth with tasks that are due the soonest (Earliest Deadline First)	Fill available bandwidth with highest priority tasks	

Scheduling Algorithms

$$\text{Tasks} \rightarrow \mathcal{P}(\text{Cond} \times \mathbb{V})$$

Deadline

Fill available bandwidth with tasks that are due the soonest (Earliest Deadline First)

Priority

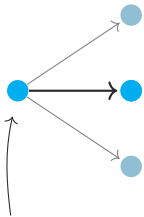
Fill available bandwidth with highest priority tasks

Priority + Deadline

Overdue tasks are assigned new, highest priority

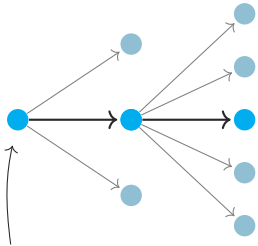
Getting stuck

Getting stuck



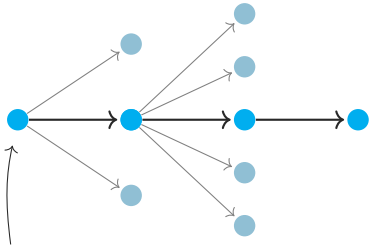
Scheduling options based
on current monitor state

Getting stuck



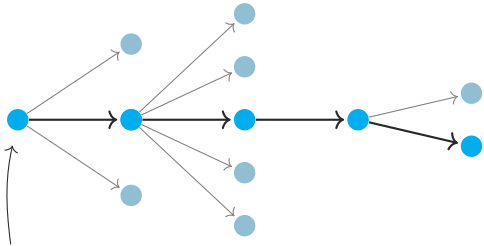
Scheduling options based
on current monitor state

Getting stuck



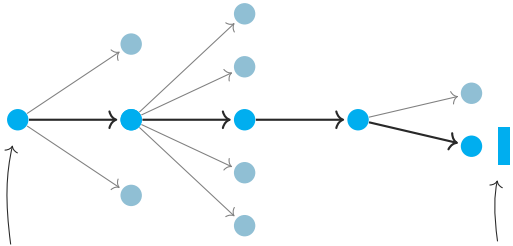
Scheduling options based
on current monitor state

Getting stuck



Scheduling options based
on current monitor state

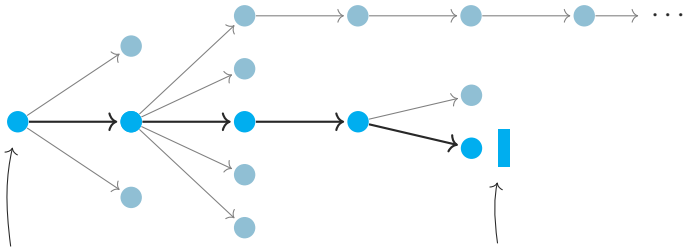
Getting stuck



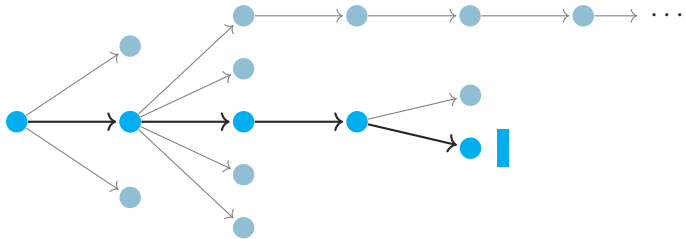
Scheduling options based on current monitor state

No option satisfies the schedule

Getting stuck



Getting stuck



A scheduling algorithm should always be able to continue, independent of the future inputs!

- **Checked by static analysis on the constraint**

Evaluation

- Can we detect triggers earlier while using the same bandwidth?

Evaluation

- Can we detect triggers earlier while using the same bandwidth?
- Can we reduce bandwidth without compromising monitoring quality?

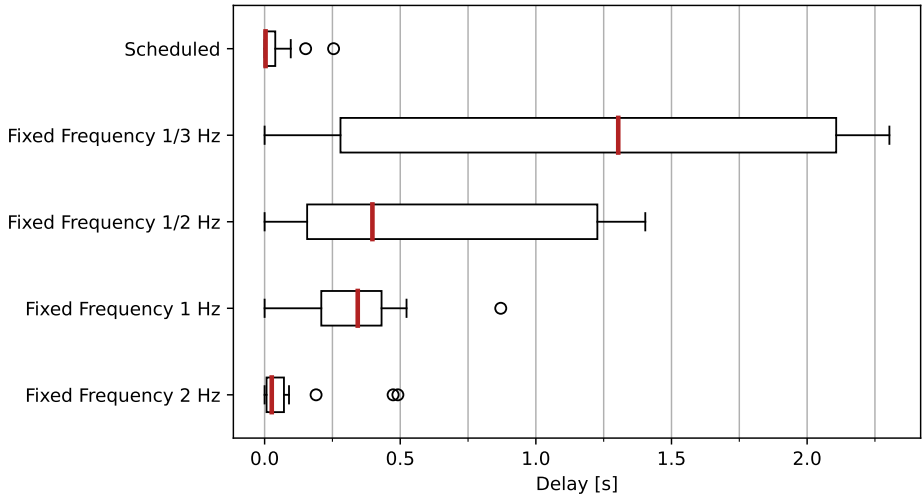
Evaluation

- Can we detect triggers earlier while using the same bandwidth?
- Can we reduce bandwidth without compromising monitoring quality?

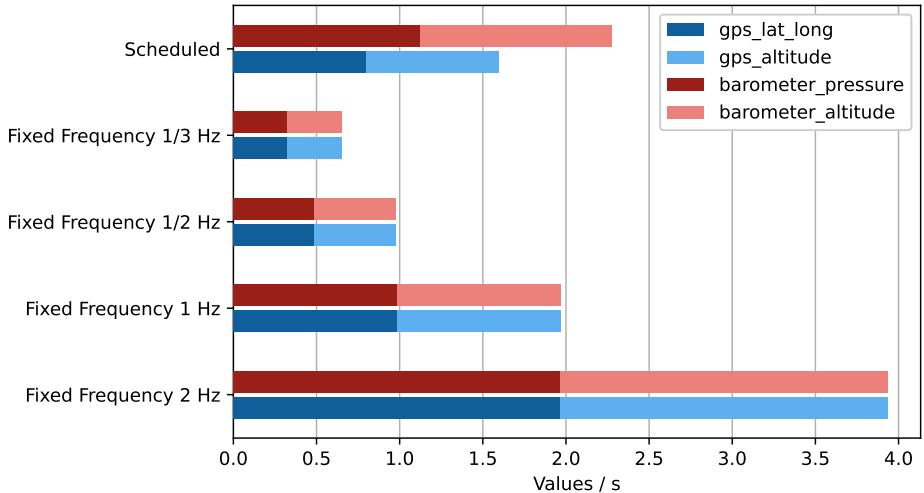
Evaluation with Microsoft AirSim [Sha+18]

- Compare scheduled monitor to multiple fixed-frequency monitors

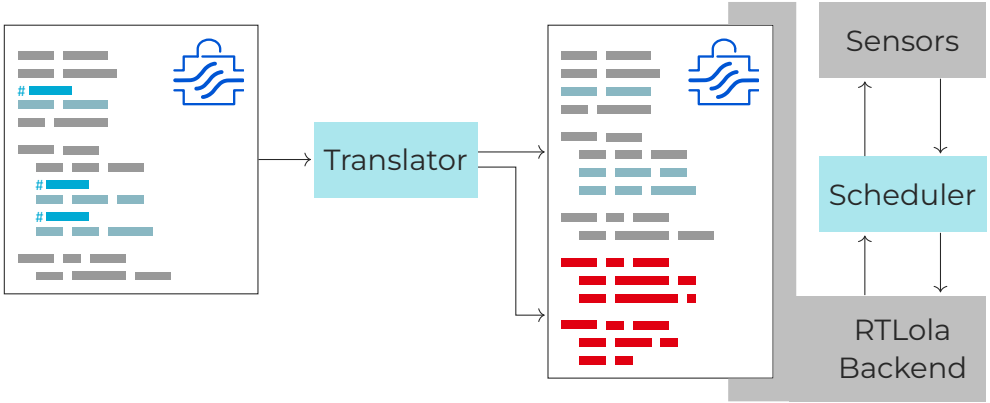
Trigger Delay Comparison



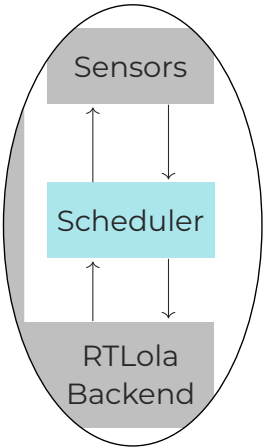
Average Bandwidth Comparison



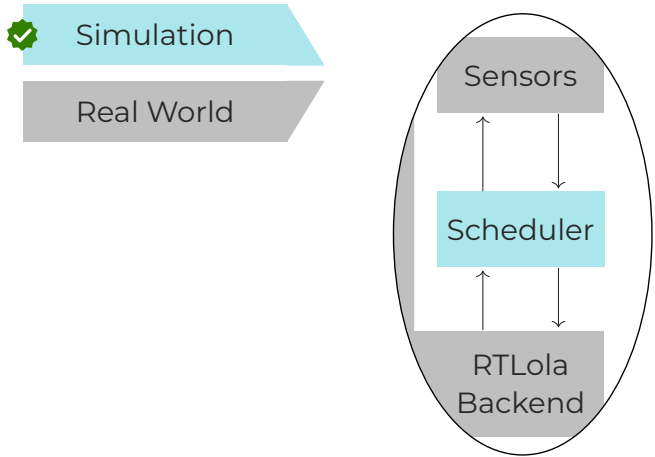
Conclusion & Outlook



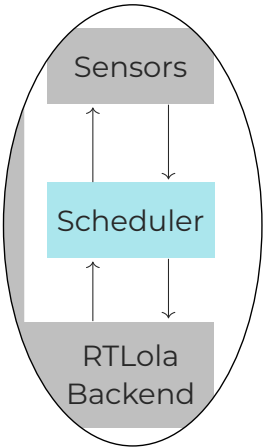
Conclusion & Outlook



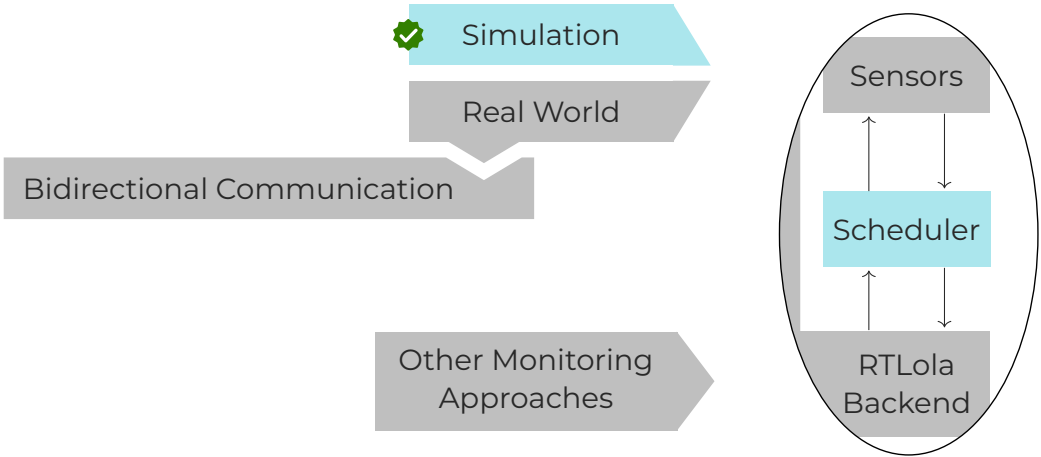
Conclusion & Outlook



Conclusion & Outlook



Conclusion & Outlook



References

[Sha+18] S. Shah, D. Dey, C. Lovett, and A. Kapoor. AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. In: Field and Service Robotics. Ed. by M. Hutter and R. Siegwart. Cham: Springer International Publishing, 2018, pp. 621–635. ISBN: 978-3-319-67361-5.