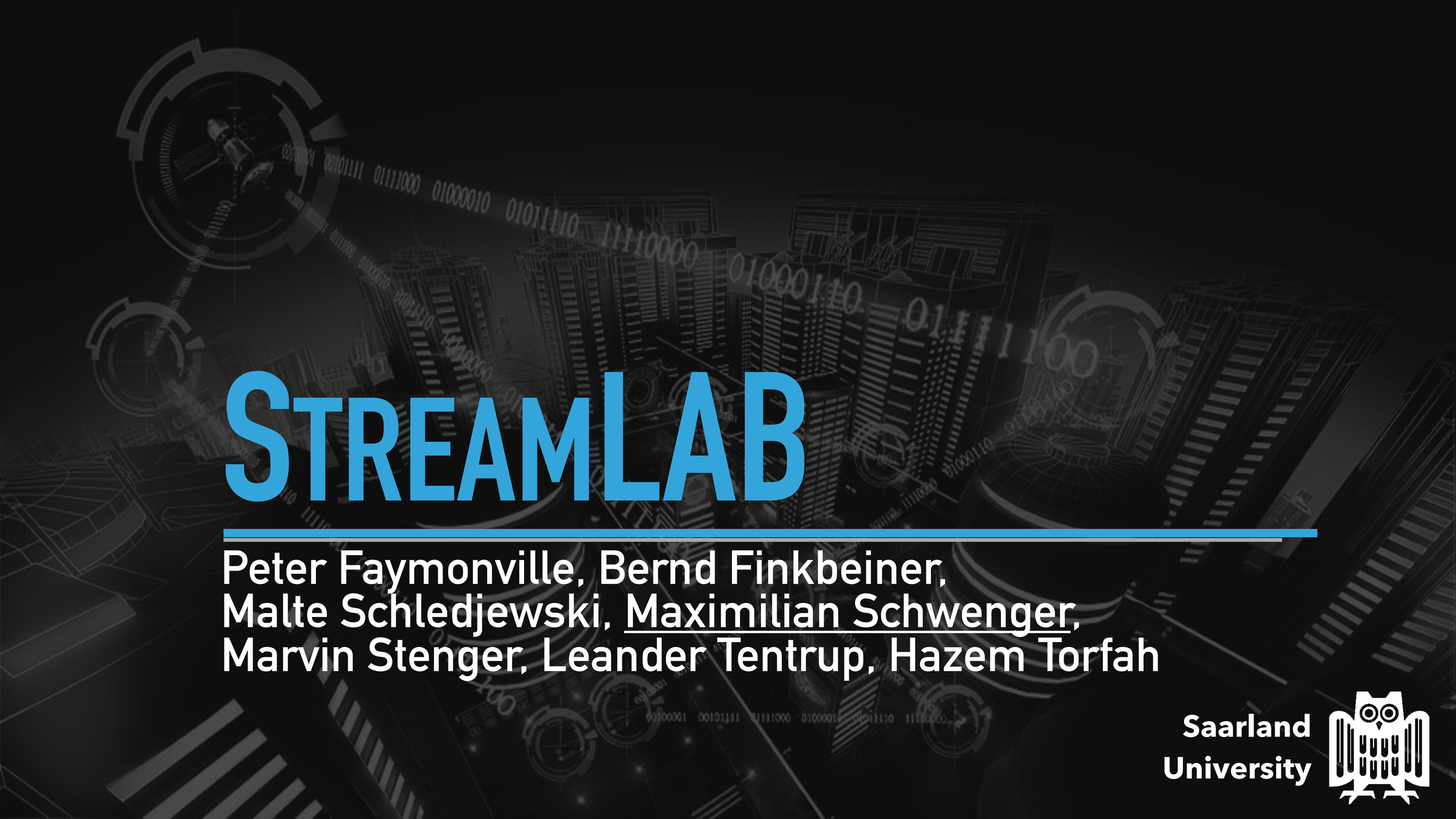
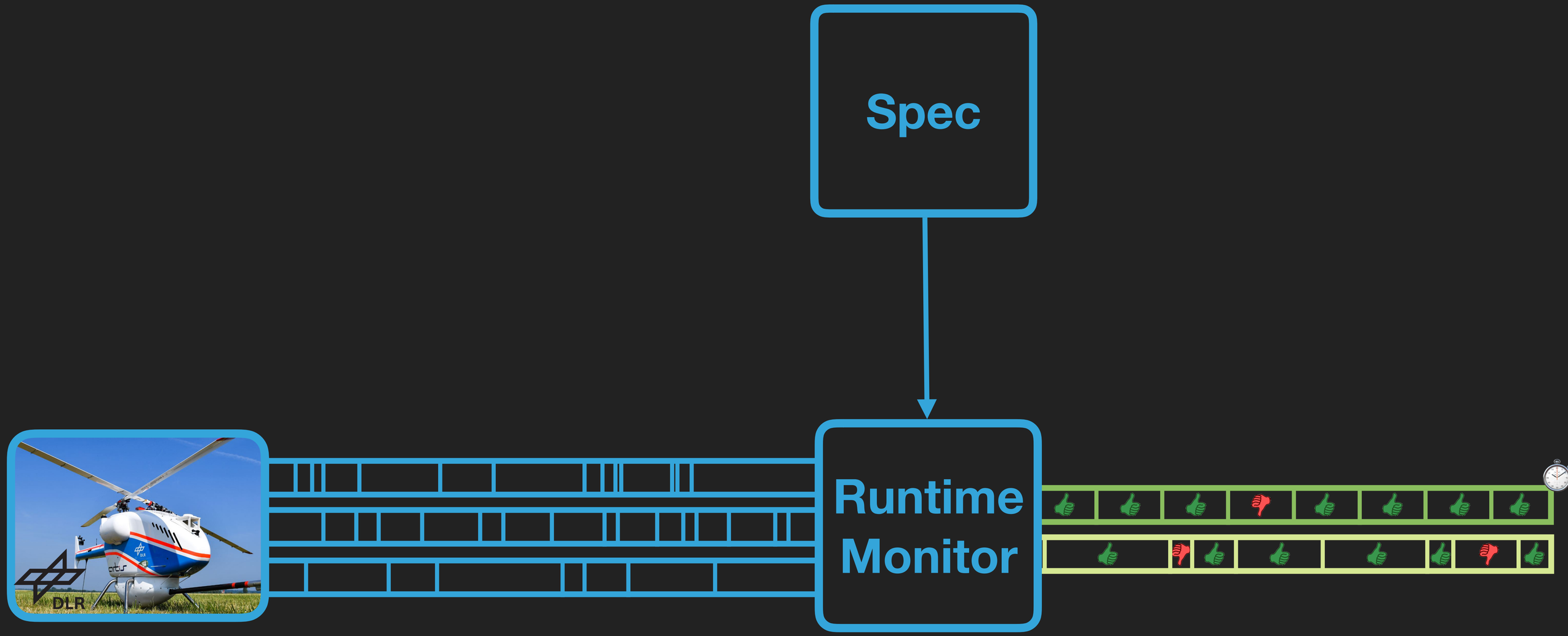


StreamLAB

Peter Faymonville, Bernd Finkbeiner,
Malte Schledjewski, Maximilian Schwenger,
Marvin Stenger, Leander Tentrup, Hazem Torfah

Saarland
University

Stream-based Monitoring



Stream-based Monitoring

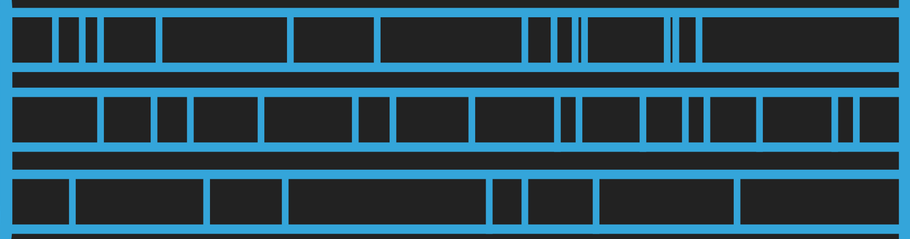
Asynchronous Events

Spec

Runtime Monitor

Stream-based Monitoring

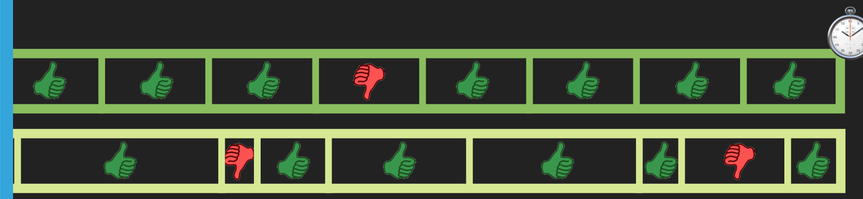
Asynchronous Events



Spec

Runtime Monitor

Static Guarantees



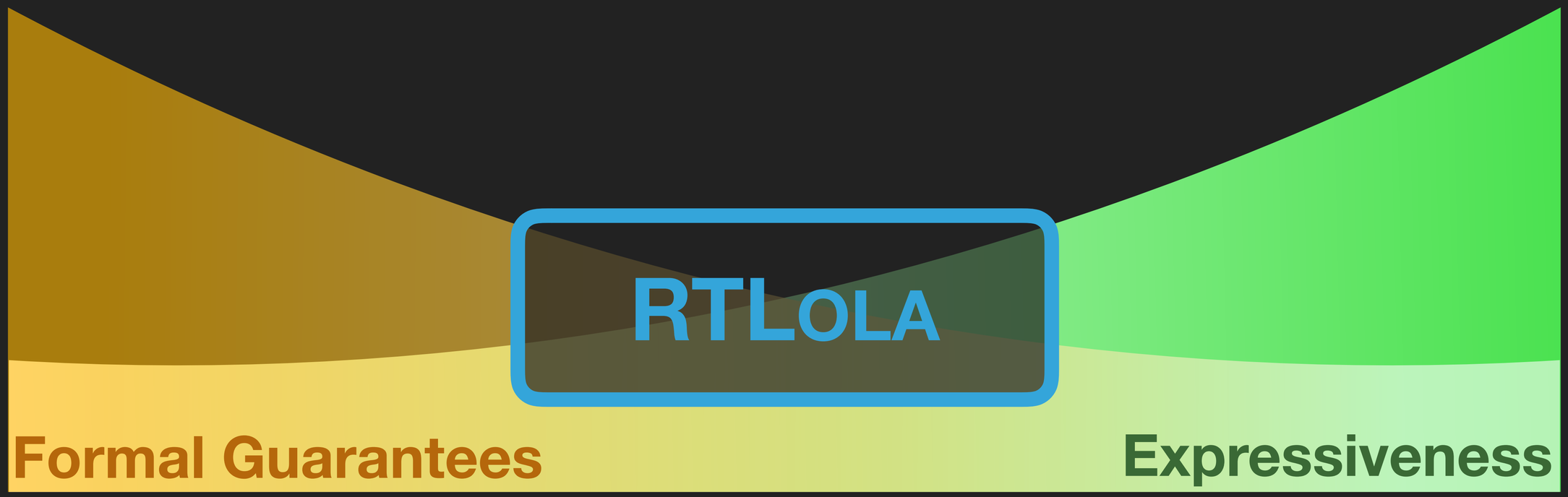
The StreamLAB Framework

RTLola Spec

Static Analyzer

Runtime Monitor

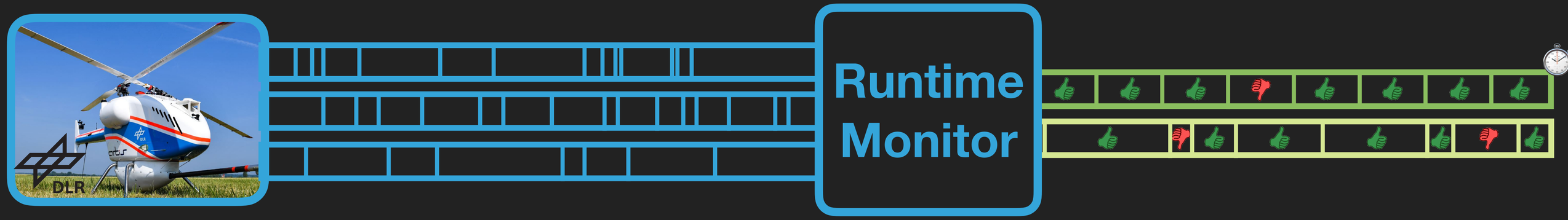
Specification Languages



RTLola in a Nutshell

```
input altitude, TAS, pitch: Float
```

Assert: Do not fly below 2000ft.

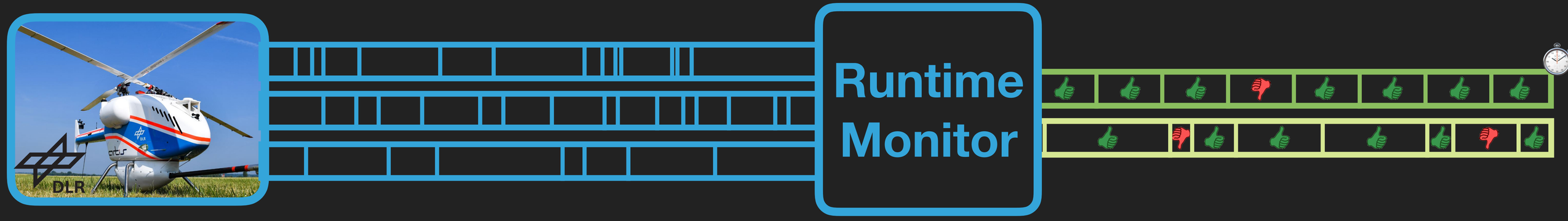


RTLola in a Nutshell

```
input altitude, TAS, pitch: Float

trigger altitude < 2000 “Flying too low.”
```

Assert: Cover at least 200 miles per hour.



RTLola in a Nutshell

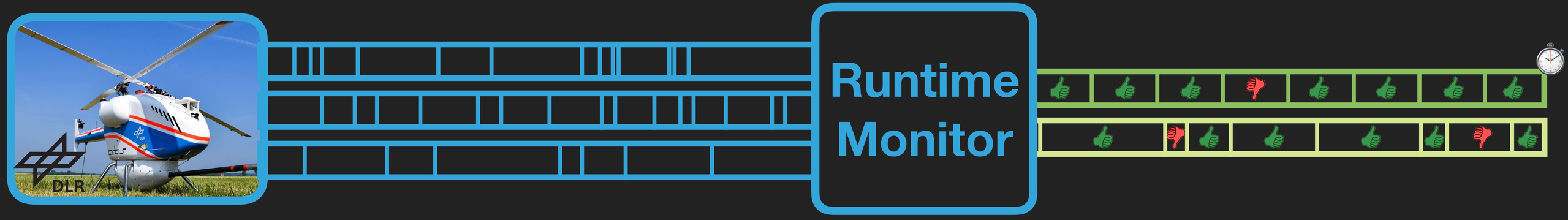
```
input altitude, TAS, pitch: Float

trigger altitude < 2000 “Flying too low.”

output gnd_spd := cos(pitch) * TAS
output gnd_dist @5Hz := gnd_spd.aggr(over: 1h, using: ∫)

trigger gnd_dist < 200 “Flying too slow.”
```

Assert: Altimeter samples with at least 10Hz.



RTLola in a Nutshell

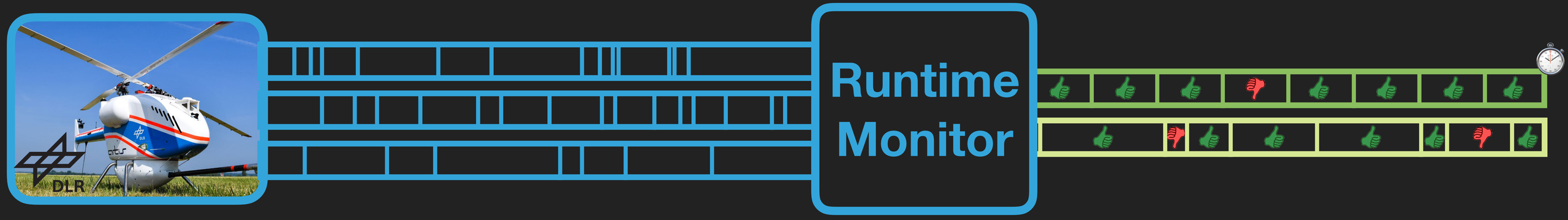
```
input altitude, TAS, pitch: Float

trigger altitude < 2000 “Flying too low.”

output gnd_spd := cos(pitch) * TAS
output gnd_dist @5Hz := gnd_spd.aggr(over: 1h, using: ∫)

trigger gnd_dist < 200 “Flying too slow.”

trigger @10Hz altitude.aggr(over: 1s, using: count) < 10 “Few samples.”
```



Experiments

Specification:

GPS frequency validation

GPS/IMU jump detection

Hover phase detection

Results:

433,000 events

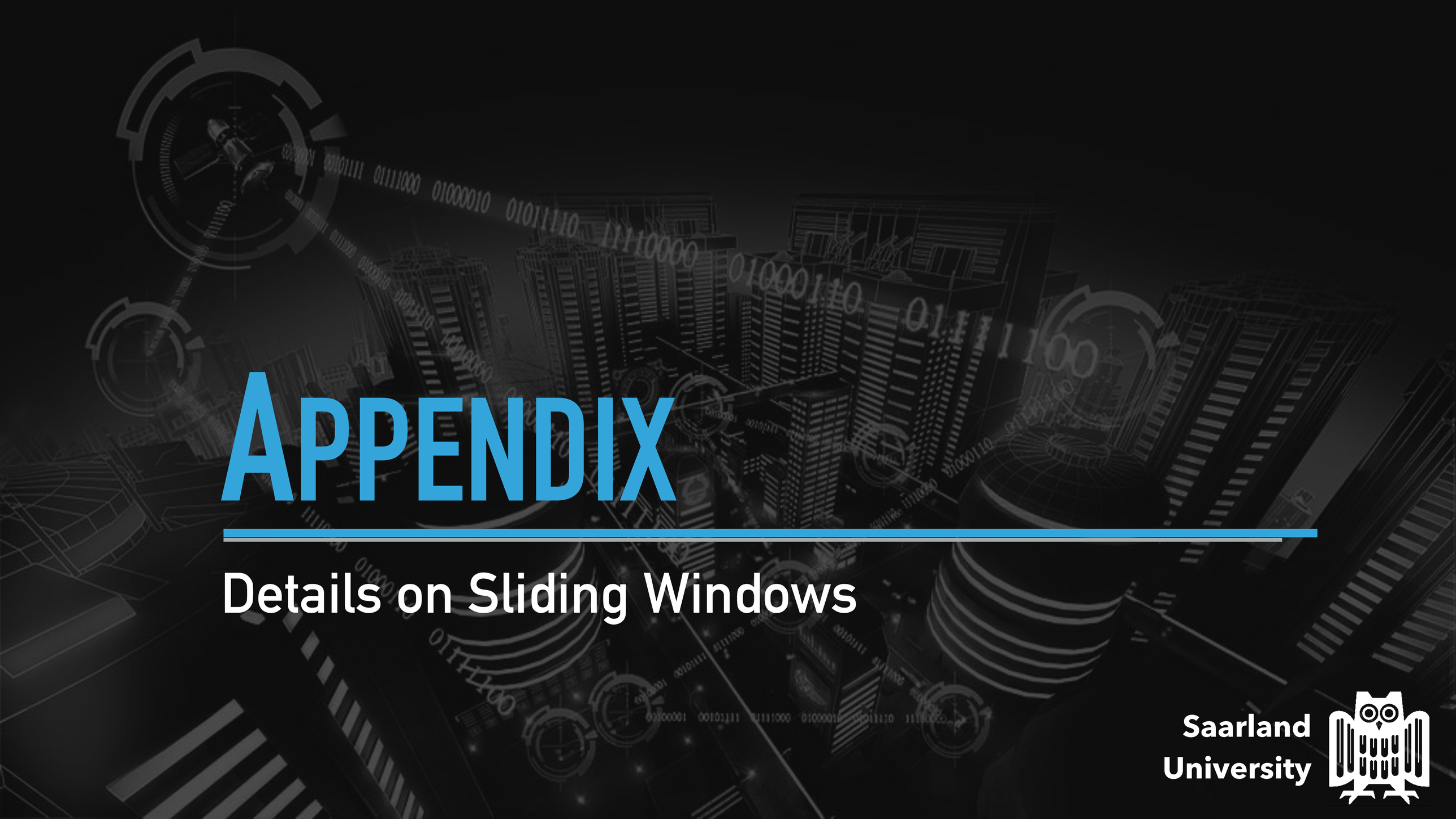
1,545ns per event @ 146%

Stack size < 1kB, no heap

Download + Tutorial:

stream-lab.eu

Contact:
stream-lab@react.uni-saarland.de



Appendix

Details on Sliding Windows

Saarland University

Appendix: List Homomorphisms

List Homomorphism:

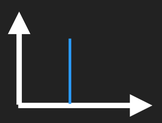
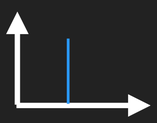
$$\gamma \to (\circ, \varepsilon, map, fin)$$

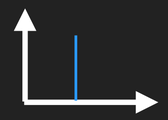
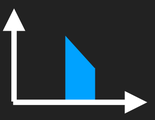
$$\gamma(a_1, \ldots, a_n) = fin(map(a_1) \circ \ldots \circ map(a_n)$$

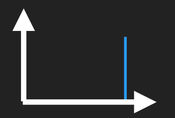
Integral:

$\varepsilon = \bot$

$map(v, ts) =$ 

$\bot \circ$  $=$  $\circ \bot =$ 

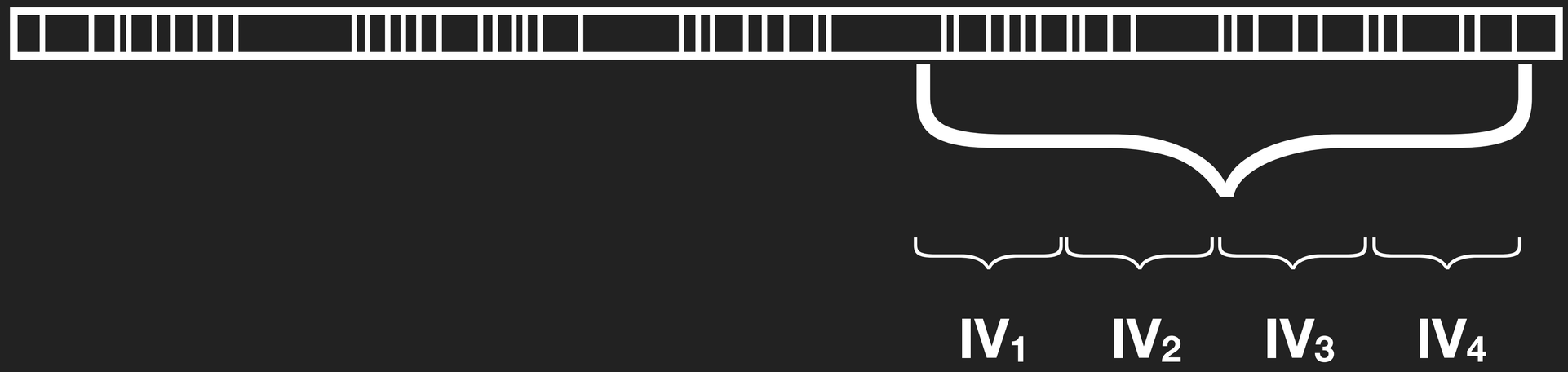
 $\circ$  $=$ 

 $\circ$  $=$ 

$fin($  $) = vol($  $)$

Meertens; *Algorithmics: Towards programming as a mathematical activity*; 1986

Appendix: Sliding Windows — Paning



Li, Maier, Tufte, Papadimos, Tucker; "*No pane, no gain: efficient evaluation of sliding-window aggregates over data streams";* SIGMOD 2005