

# ON THE SIMILARITIES OF AIRCRAFT AND HUMANS: MONITORING CYBER-PHYSICAL SYSTEMS WITH STREAMLAB

---

**Jan Baumeister**, Bernd Finkbeiner,  
Maximilian Schwenger, Hazem Torfah

# ON THE SIMILARITIES OF AIRCRAFT AND HUMANS

## StreamLAB

**Runtime  
Monitor**

**RTLola**



**Static  
Analyzer**

# STATIC VERIFICATION

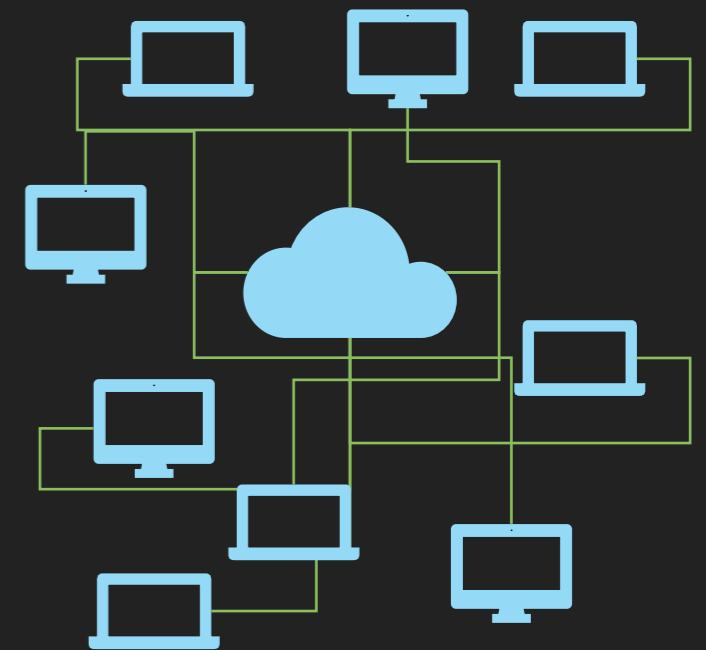
## Lack of Knowledge



## Complexity

$$\begin{aligned}\dot{p} &= Rv \\ \dot{R} &= R\hat{\omega} \\ \dot{v} &= -\omega \times v + R^T \bar{g} + \\ &\quad f_v(\omega, v, \alpha, \beta, \omega_r, \delta_c, \delta_r) \\ \dot{\omega} &= -J^{-1}(\omega \times J\omega) + \\ &\quad f_w(\omega, v, \alpha, \beta, \omega_r, \delta_c, \delta_r) \\ \dot{\alpha} &= f_\alpha(\omega, v, \alpha, \beta, \omega_r, \delta_a, \delta_e) \\ \dot{\beta} &= f_\beta(\omega, v, \alpha, \beta, \omega_r, \delta_a, \delta_e) \\ \dot{\omega}_r &= f_r(\omega, v, \omega_r, \delta_c, \delta_r)\end{aligned}$$

## Non-Determinism



VERIFY:

$$\forall \sigma \in \text{runs}(S \parallel C): \sigma \models \varphi$$

# RUNTIME MONITORING

---

VERIFY:

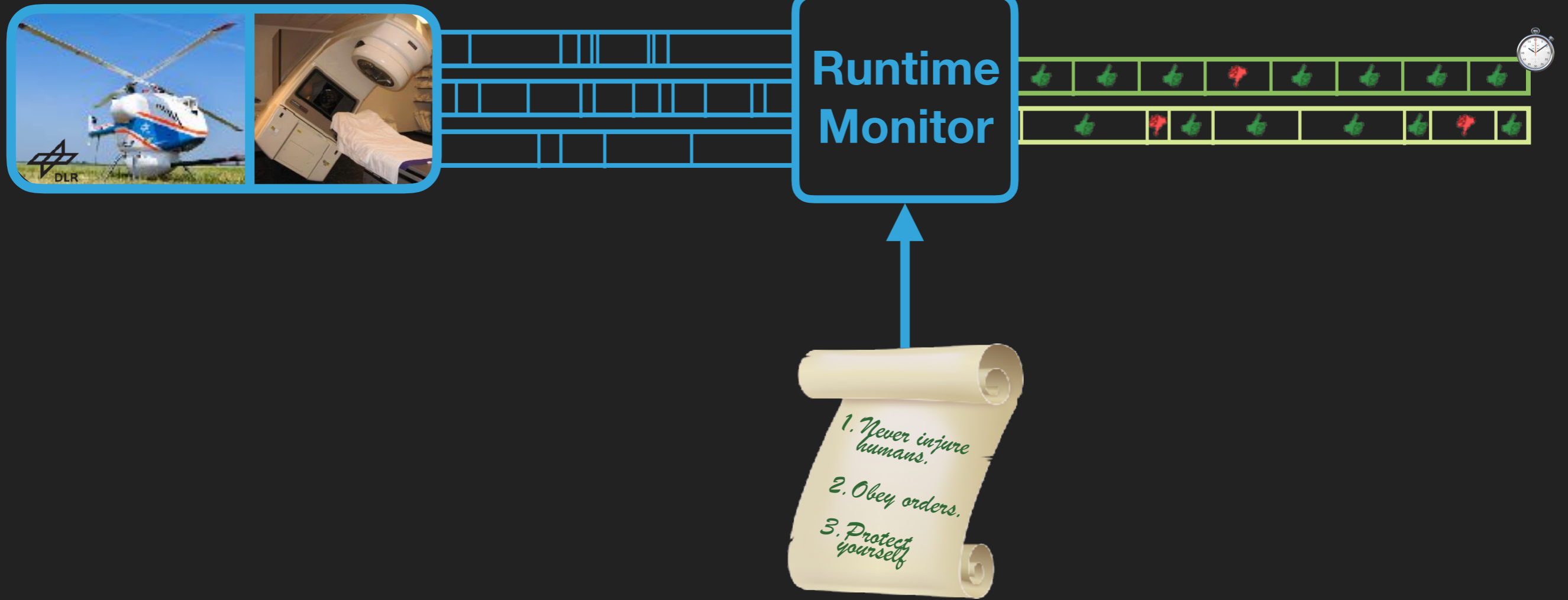
$\forall \sigma \in \text{runs}(S \parallel C): \sigma \models \varphi$



Given  $\sigma \in \text{runs}(S \parallel C):$

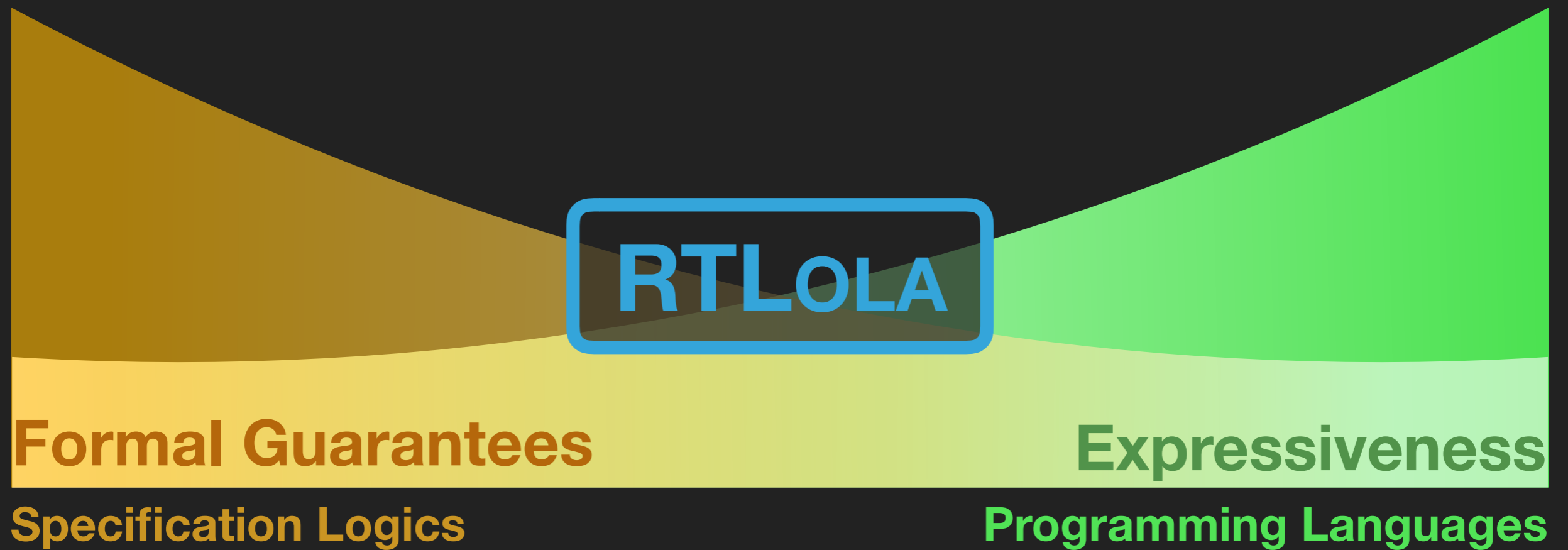
$\sigma \models \varphi$

# STREAM-BASED MONITORING



# TRADE-OFF

---



# FIRST EXAMPLE

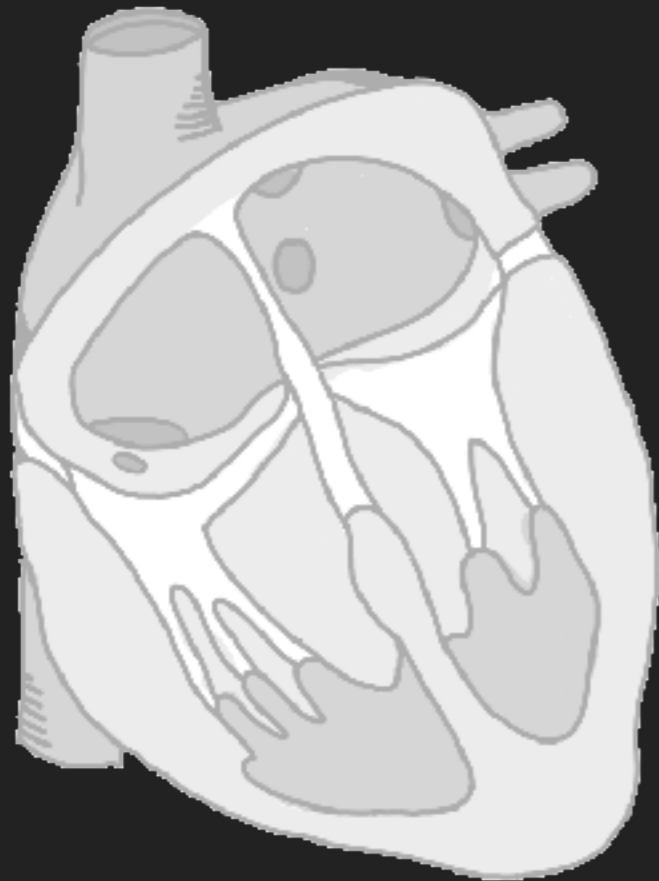
---

`input` altitude: Int64

`output` height := altitude + 6

`trigger` height > 100 "drone does not fit in the tunnel"

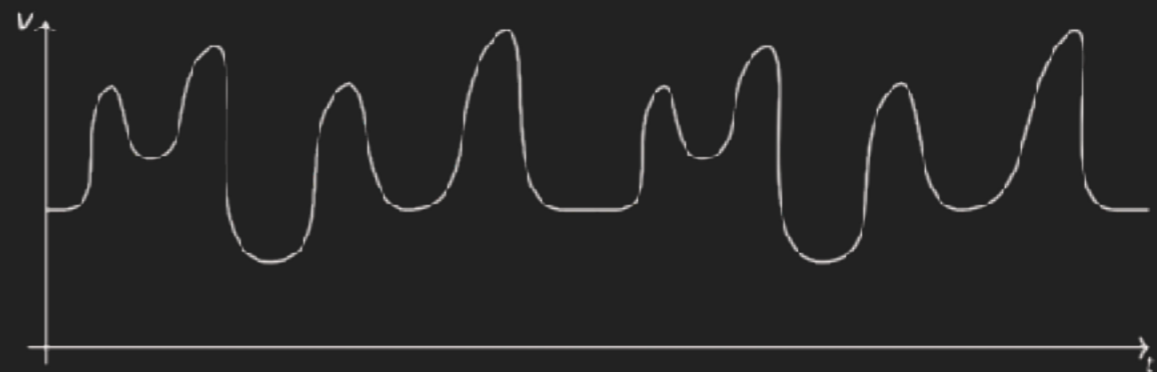
# SINUSRYTHM



## Sinusrythm



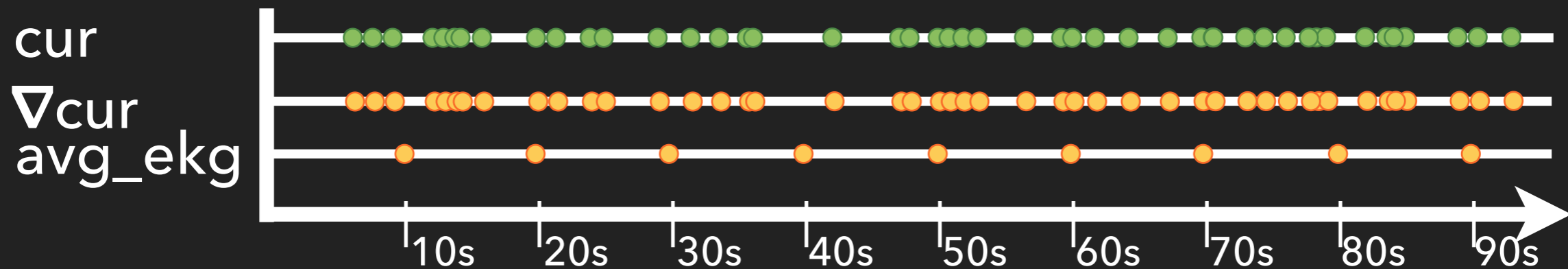
## Ventricular Fibrillation





# VENTRICULAR FIBRILLATION

```
input cur: Float64
output avg_ekg @0.1Hz := cur.aggregate(over: 10min, using: avg)
output ∇cur := derive(1, cur)
output avg_der @0.1Hz := ∇cur.aggregate(over: 10min, using: avg)
output peaking
  := cur > avg_ekg.hold() + ε1 ∧ ∇cur > avg_der.hold() + ε2
output r_peak := peaking ∧ ¬ peaking.offset(by: -1)
output heart_rate @0.1Hz := r_peak.aggregate(over: 1min, using: Σ)
trigger 200 < heart_rate < 350    "ventricular flutter"
trigger heart_rate ≥ 350    "ventricular fibrillation"
```



# GUARANTEES

---

Formal  
Semantics

Memory  
Bound

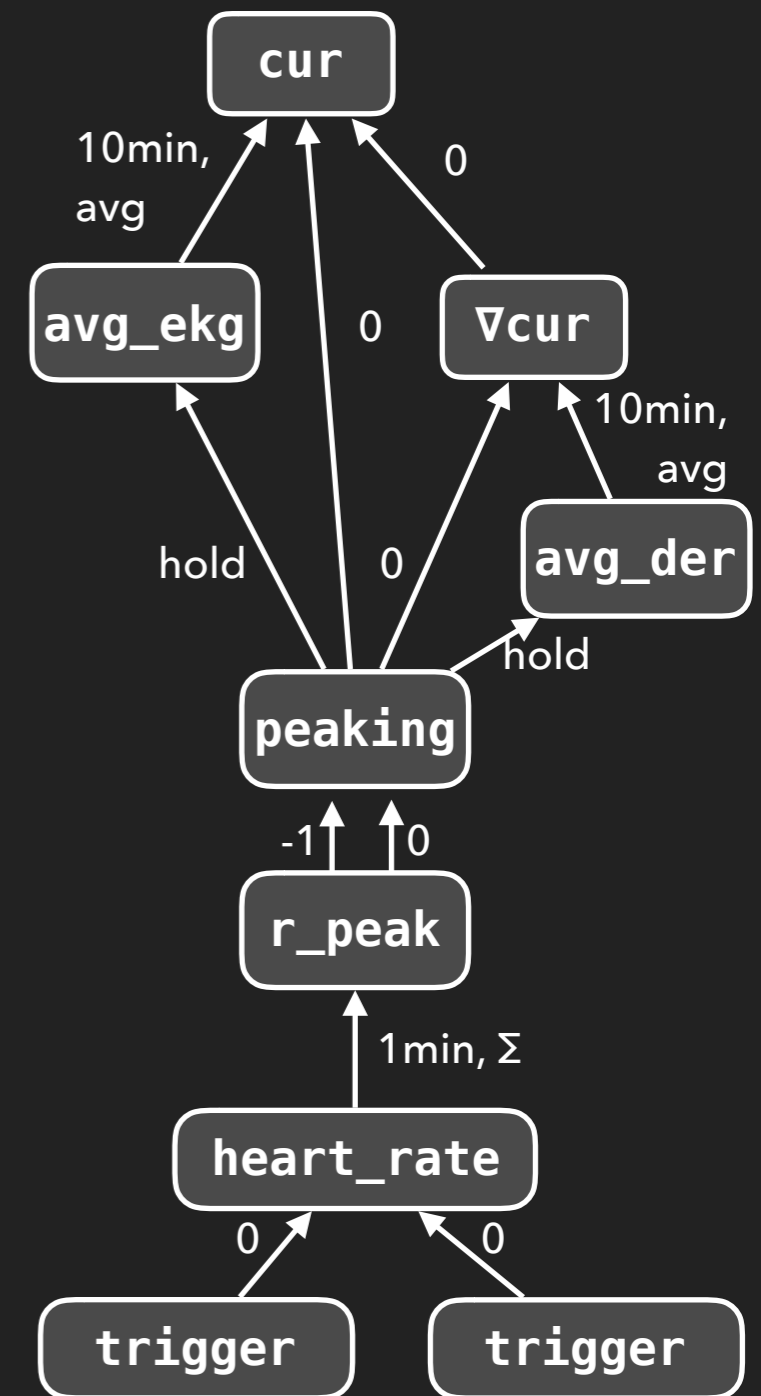
Termination  
Guarantee

Type  
System

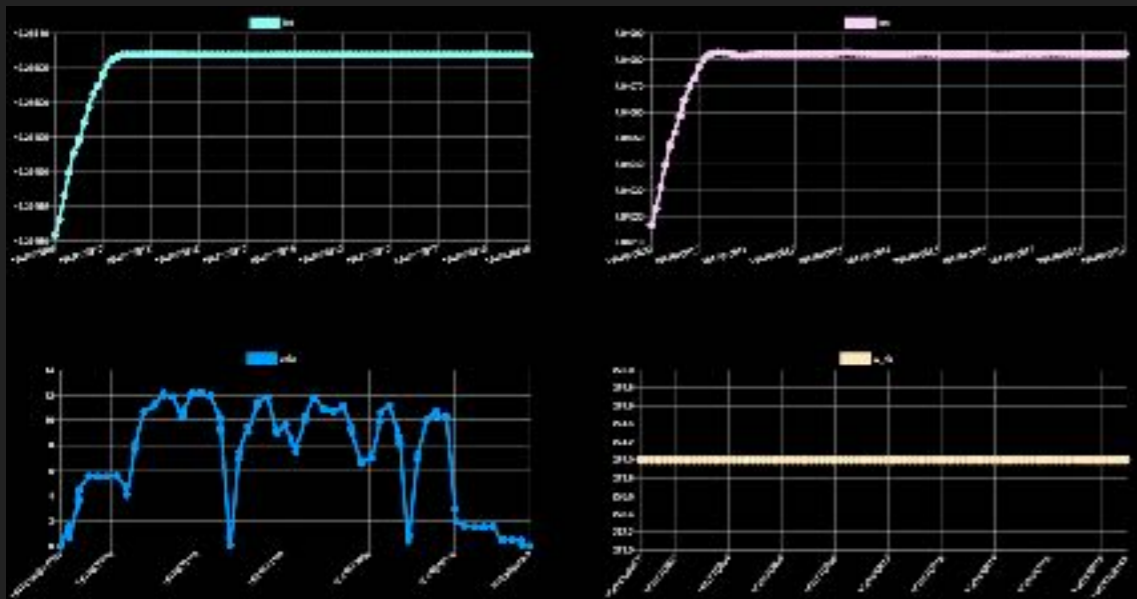
# GUARANTEES

```

input cur: Float64 Float64 @ {cur}
output  $\nabla$ cur := derive(1, cur) Float64 @ {cur}
output avg_ekg @0.1Hz Float64 @ {0.1Hz}
    := cur.aggregate(over: 10min, using: avg)
output avg_der @0.1Hz Float64 @ {0.1Hz}
    :=  $\nabla$ cur.aggregate(over: 10min, using: avg)
output peaking Bool @ {cur}
    := cur > avg_ekg.hold() +  $\epsilon_1$ 
     $\wedge$   $\nabla$ cur > avg_der.hold() +  $\epsilon_2$ 
output r_peak := peaking Bool @ {cur}
     $\wedge$   $\neg$  peaking.offset(by: -1)
output heart_rate @0.1Hz Int64 @ {0.1Hz}
    := r_peak.aggregate(over: 1min, using:  $\Sigma$ )
trigger Bool @ {0.1Hz}
    200 < heart_rate < 350 "ventricular flutter"
trigger Bool @ {0.1Hz}
    heart_rate  $\geq$  350 "ventricular fibrillation"
  
```

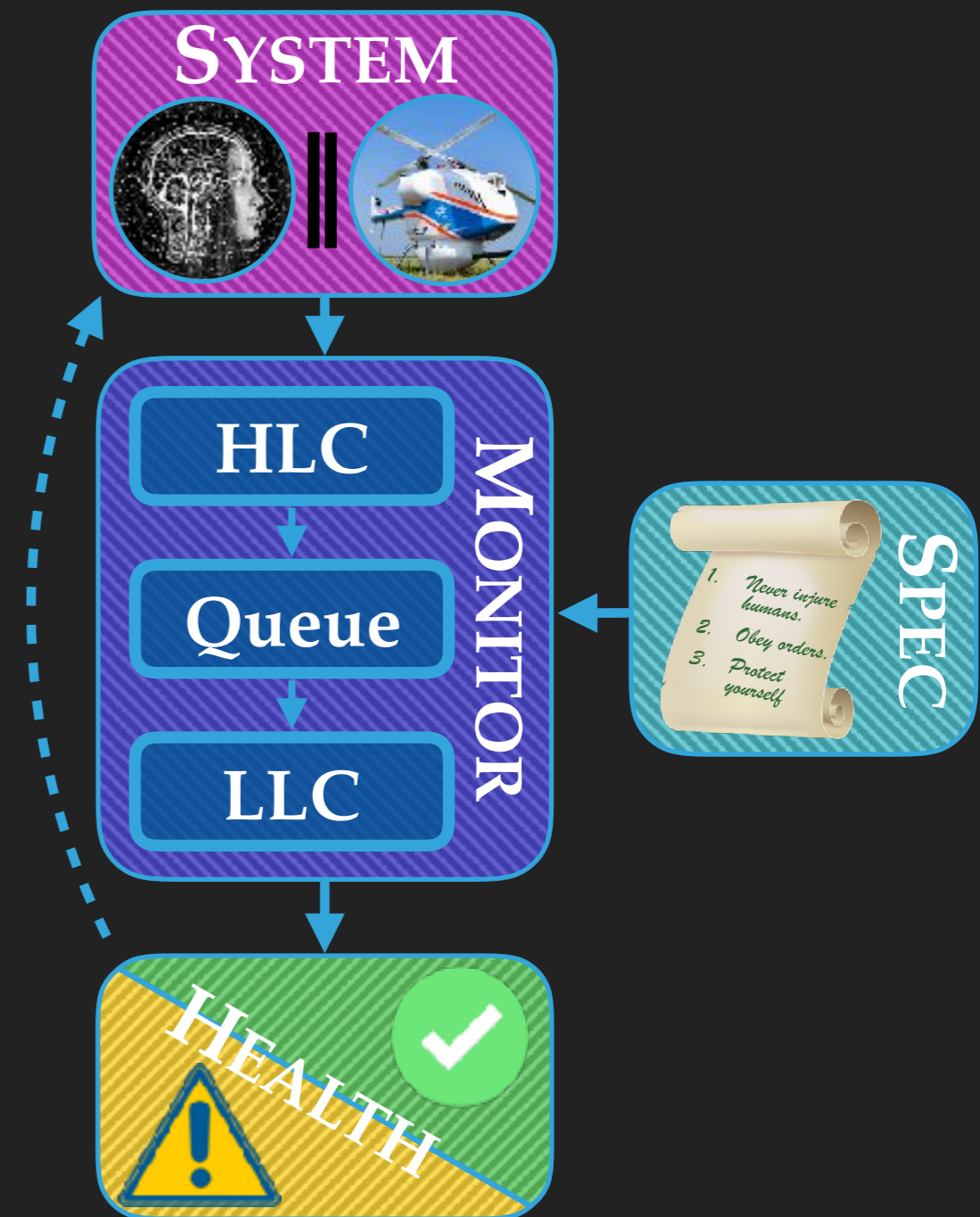


# INTERPRETER



Peter Faymonville, Bernd Finkbeiner, Malte Schledjewski, Maximilian Schwenger, Marvin Stenger, Leander Tentrup, and Hazem Torfah. *StreamLAB: Stream-based Monitoring of Cyber-physical Systems* (CAV 2019)

# VHDL COMPILATION



Jan Baumeister, Bernd Finkbeiner, Maximilian Schwenger, and Hazem Torfah. *FPGA Stream-Monitoring of Real-time Properties* (EMSOFT 2019)

# CONCLUSION

**StreamLAB**

[www.stream-lab.eu](http://www.stream-lab.eu)

