FPGA Stream-Monitoring of Real-Time Properties

Jan Baumeister, Bernd Finkbeiner,
Maximilian Schwenger, Hazem Torfah

EMBEDDED SYSTEMS WEEK

CPEC CENTER FOR PERSPICUOUS COMPUTING

OSARES

erc

Saarland University

Stream Monitoring

Output Streams

Verdict

Verdict

Runtime Monitor

Spec

Stream Monitoring

Output Streams

Verdict

Verdict

FPGA Runtime Monitor

VHDL

Spec

Stream Monitoring

Output Streams

Verdict

Verdict

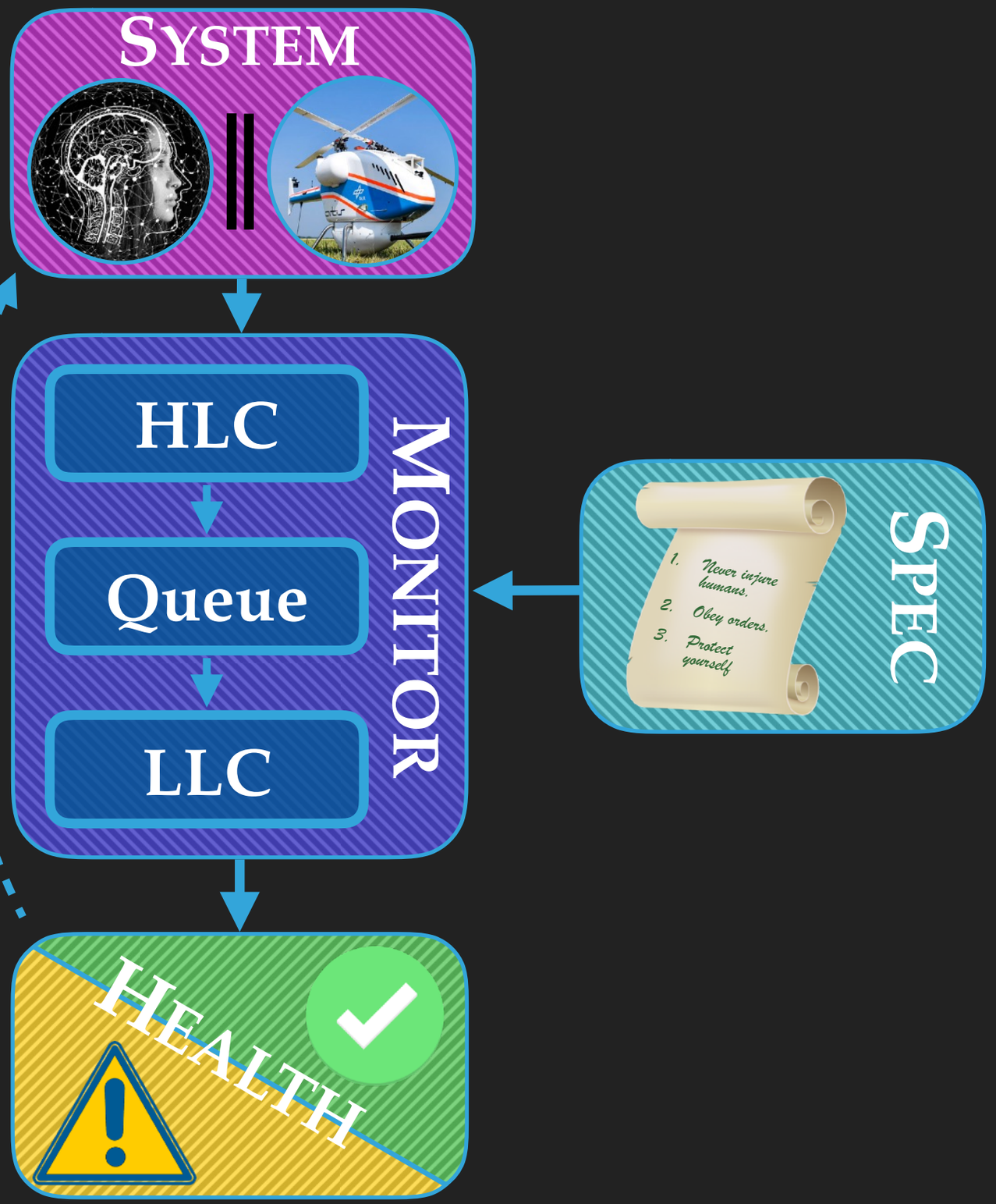
FPGA Runtime Monitor

- Easy to Integrate
- Low Resource Consumption
- Formal Guarantees

VHDL

Spec

Overview

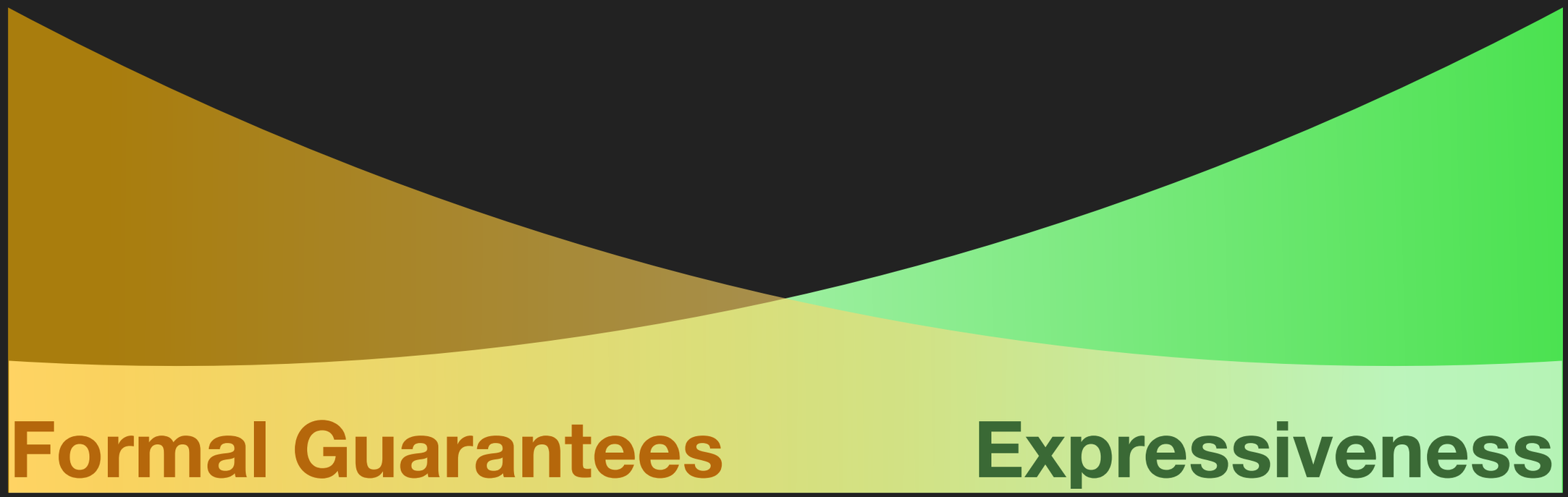

System
HLC
Queue
LLC
Monitor
Spec
1. Never injure humans.
2. Obey orders.
3. Protect yourself
Health

Specification Language

Formal Guarantees

Specification Logics

Specification Language



Specification Logics **Programming Languages**

Specification Language

RTLola in a Nutshell

```
input lat, lon, velo: Float64
```

The GPS module operates with at least 5Hz.

RTLola in a Nutshell

```
input lat, lon, velo: Float64


output gps_freq @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5 “GPS frequency less than 5 Hz.”
```

RTLola in a Nutshell

```
input lat, lon, velo: Float64


output gps_freq @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5 “GPS frequency less than 5 Hz.”
```

Measured and computed velocities coincide.

RTLola in a Nutshell

```
input lat, lon, velo: Float64

output gps_freq @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5 “GPS frequency less than 5 Hz.”

output gps_dist := sqrt(δ(lon)² + δ(lat)²)
output gps_velo := ∇(gps_dist)
trigger abs(gps_velo - velo) > 0.1 “Conflicting measurements for velocity.”
```

RTLola in a Nutshell

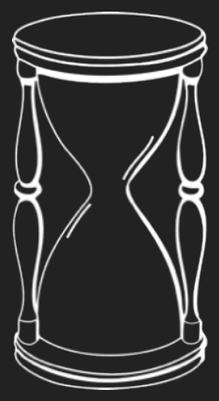
```
input lat, lon, velo: Float64

output gps_freq @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5 “GPS frequency less than 5 Hz.”

output gps_dist := sqrt(δ(lon)² + δ(lat)²)
output gps_velo := ∇(gps_dist)
trigger abs(gps_velo - velo) > 0.1 “Conflicting measurements for velocity.”
```

Deceleration is preceded by a slow-down command.

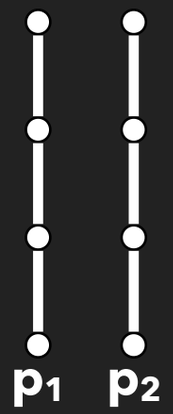
RTLola in a Nutshell

```
input lat, lon, velo: Float64
input slow_down_cmd: Bool

output gps_freq @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5 “GPS frequency less than 5 Hz.”

output gps_dist := sqrt(δ(lon)² + δ(lat)²)
output gps_velo := ∇(gps_dist)
trigger abs(gps_velo - velo) > 0.1 “Conflicting measurements for velocity.”

output fast := velo > 700
output slow_down := fast.offset(by: -1).defaults(to: false) ∧ ¬fast
trigger @1Hz ¬slow_down_cmd.aggregate(over: 5s, using: ∃)
                ∧ slow_down.hold().defaults(to: false) “Spurious Slow-Down.”
```

Challenges



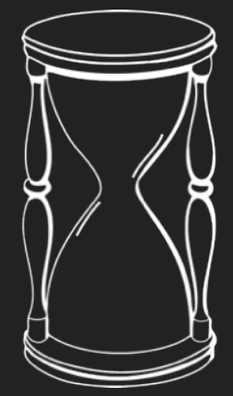
Periodic versus Event-Based



Utilize Parallel Execution

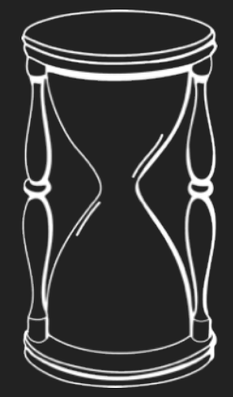
Reduce Circuit Cost

Periodic v. Event-Based



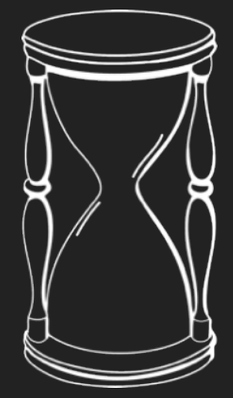
```
input lat, lon, velo: Float64
input slow_down_cmd: Bool

output gps_freq @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5 “GPS frequency less than 5 Hz.”

output gps_dist := sqrt(δ(lon)² + δ(lat)²)
output gps_velo := ∇(gps_dist)
trigger abs(gps_velo - velo) > 0.1 “Conflicting measurements for velocity.”
…
```

Periodic v. Event-Based

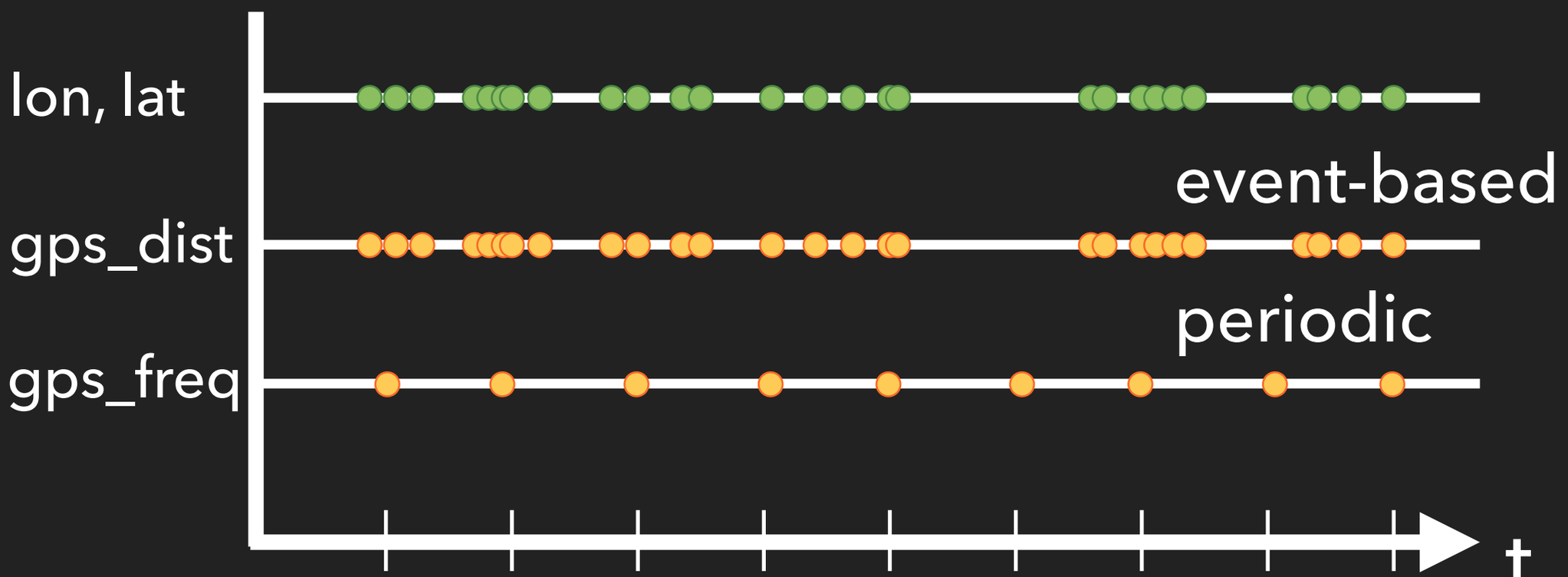


```
input lat, lon, velo: Float64
input slow_down_cmd: Bool

output gps_freq @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5 "GPS frequency less than 5 Hz."

output gps_dist := sqrt(δ(lon)² + δ(lat)²)
output gps_velo := ∇(gps_dist)
trigger abs(gps_velo - velo) > 0.1 "Conflicting measurements for velocity."
...
```

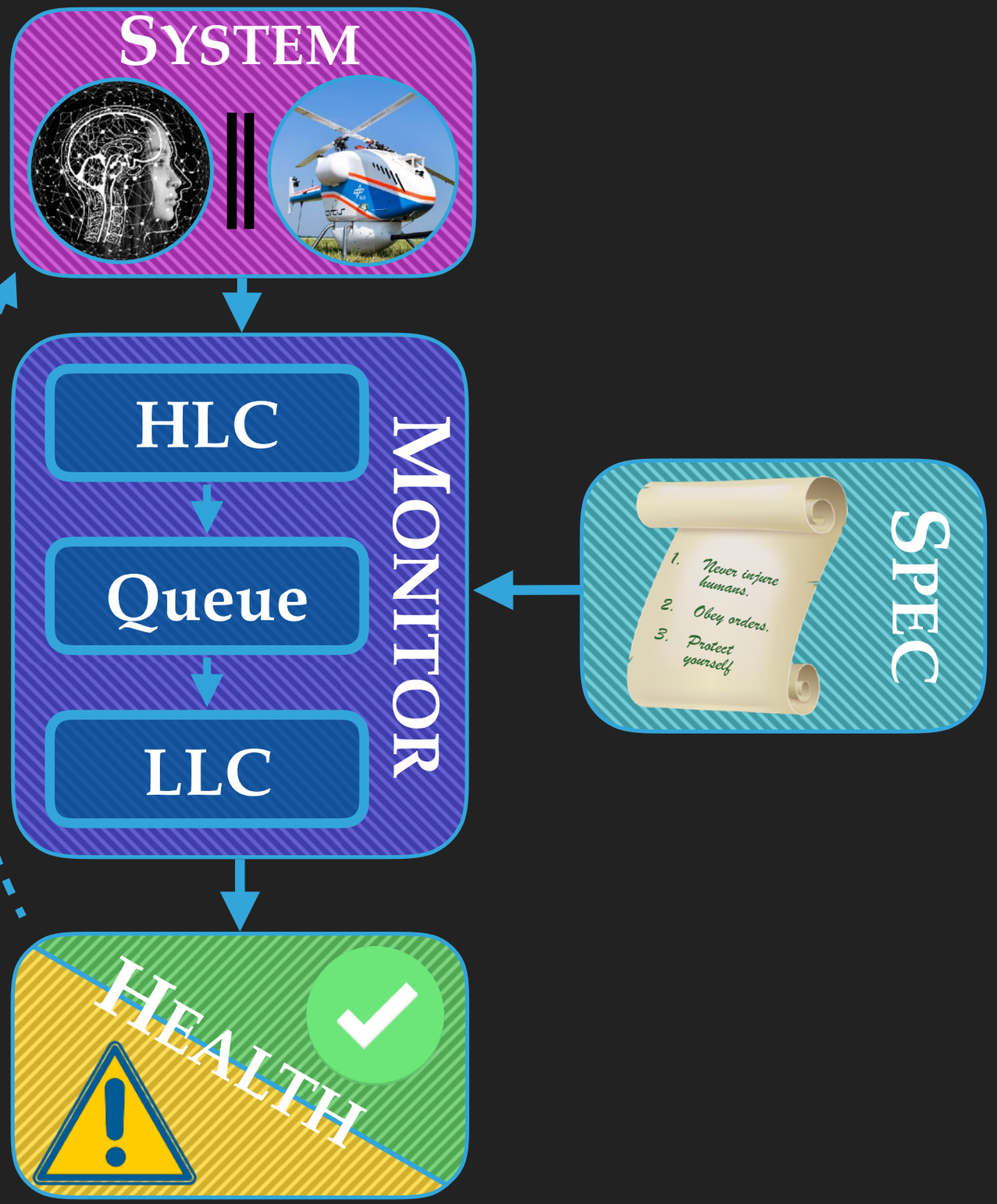
Periodic v. Event-Based



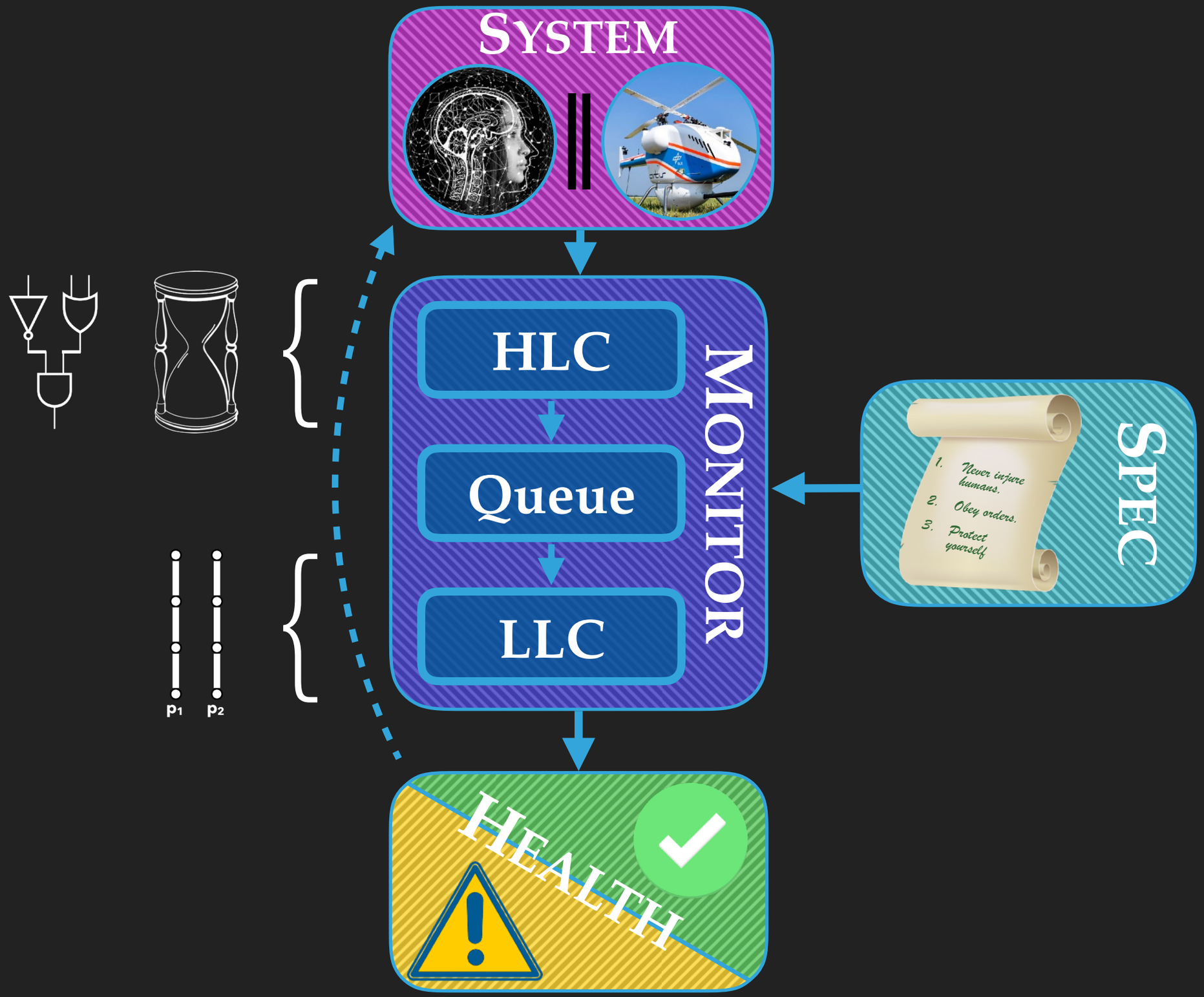
```
input lat, lon, velo: Float64
input slow_down_cmd: Bool

output gps_freq @1Hz := lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5 "GPS frequency less than 5 Hz."

output gps_dist := sqrt(δ(lon)² + δ(lat)²)
output gps_velo := ∇(gps_dist)
trigger abs(gps_velo - velo) > 0.1 "Conflicting measurements for velocity."
...
```



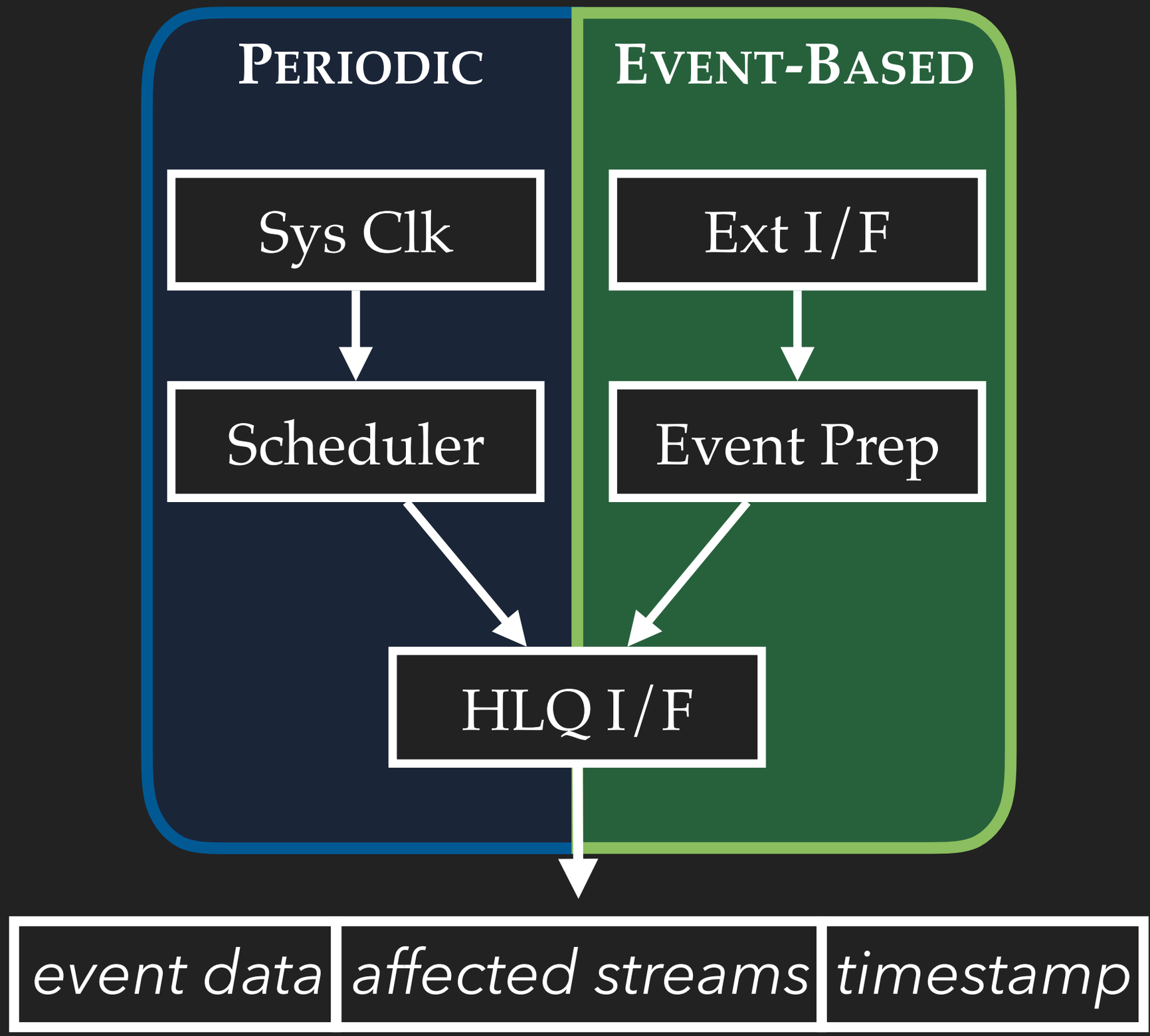
From RTLola to VHDL



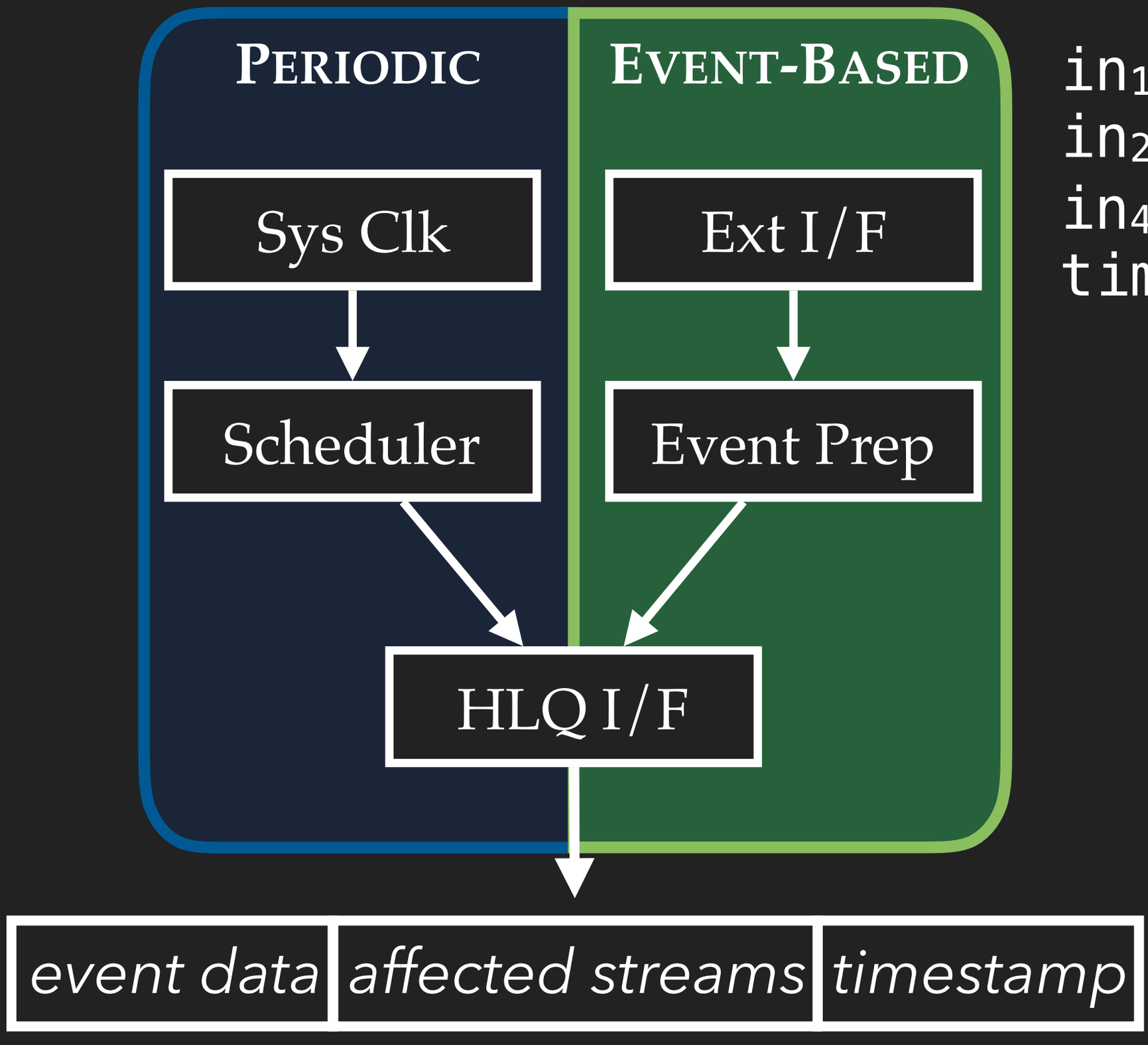
From RTLola to VHDL



High-Level Controller

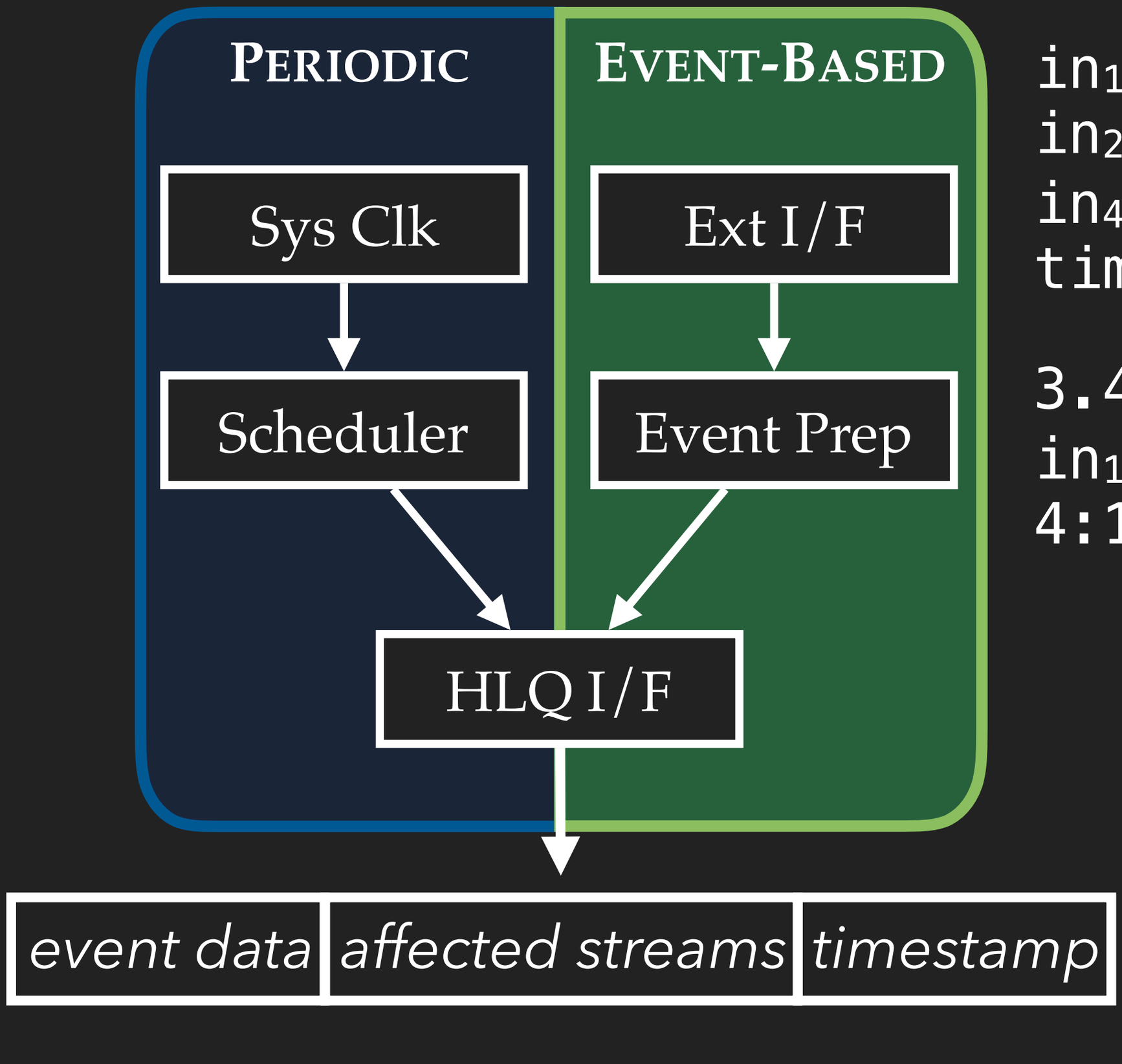


High-Level Controller



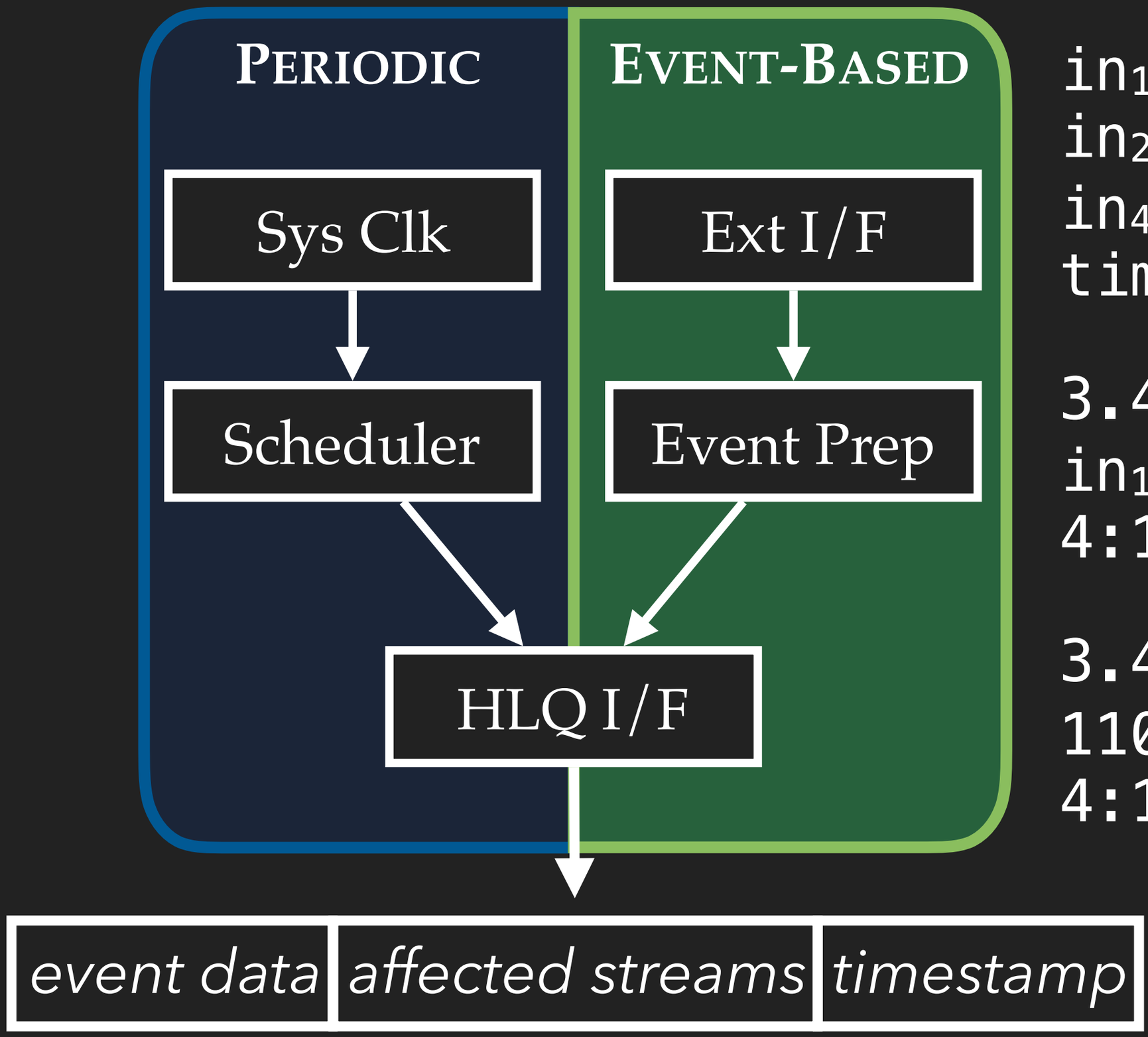
in_1 → 3.4
in_2 → T
in_4 → 9
time → 4:15pm

High-Level Controller




```
in1 → 3.4
in2 → T
in4 → 9
time → 4:15pm

3.4,T,#,9;
in1,in2,in4,out2,…
4:15pm
```

High-Level Controller

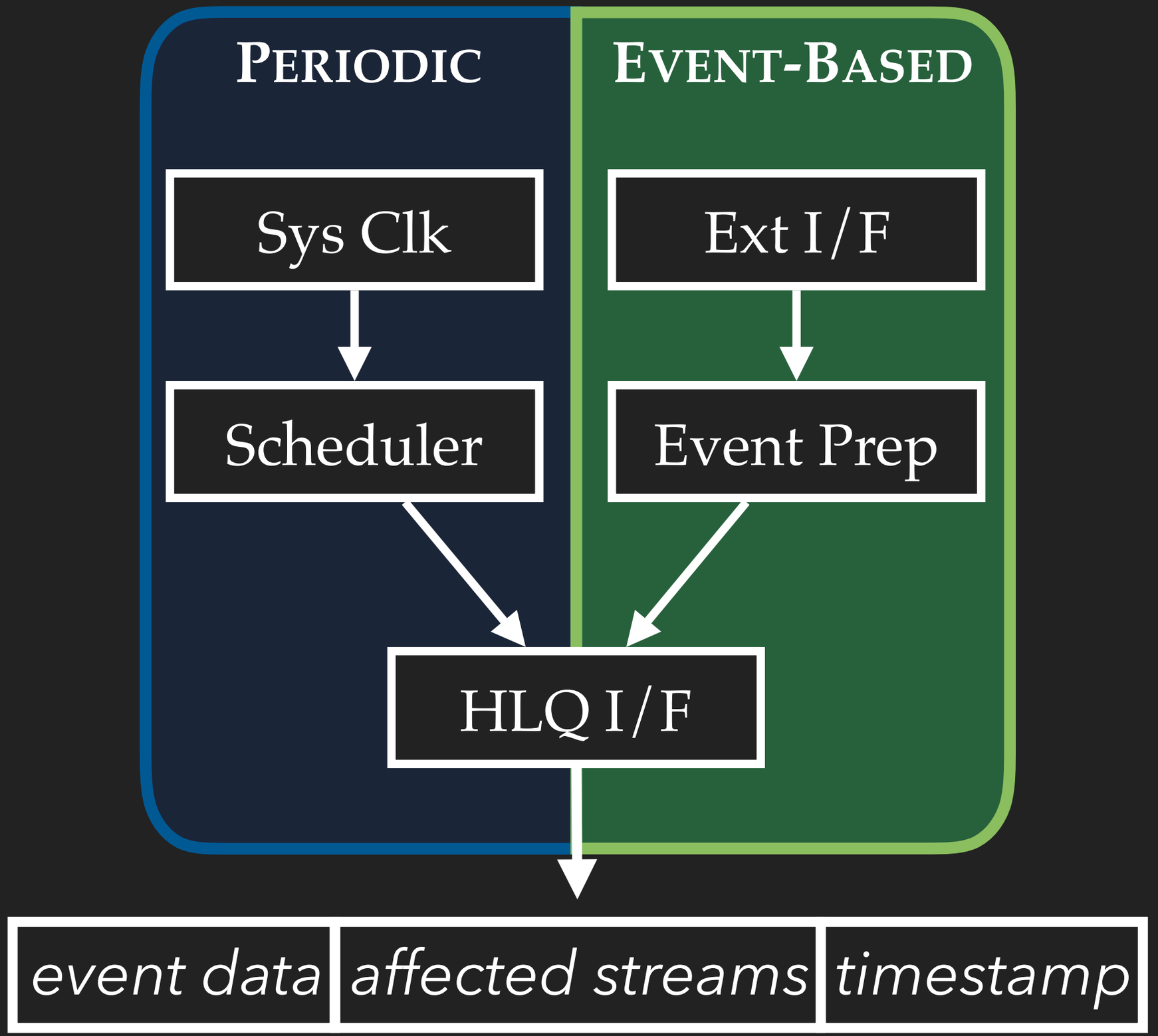


in_1 → 3.4
in_2 → T
in_4 → 9
time → 4:15pm

3.4,T,#,9;
in_1,in_2,in_4,out_2,…
4:15pm

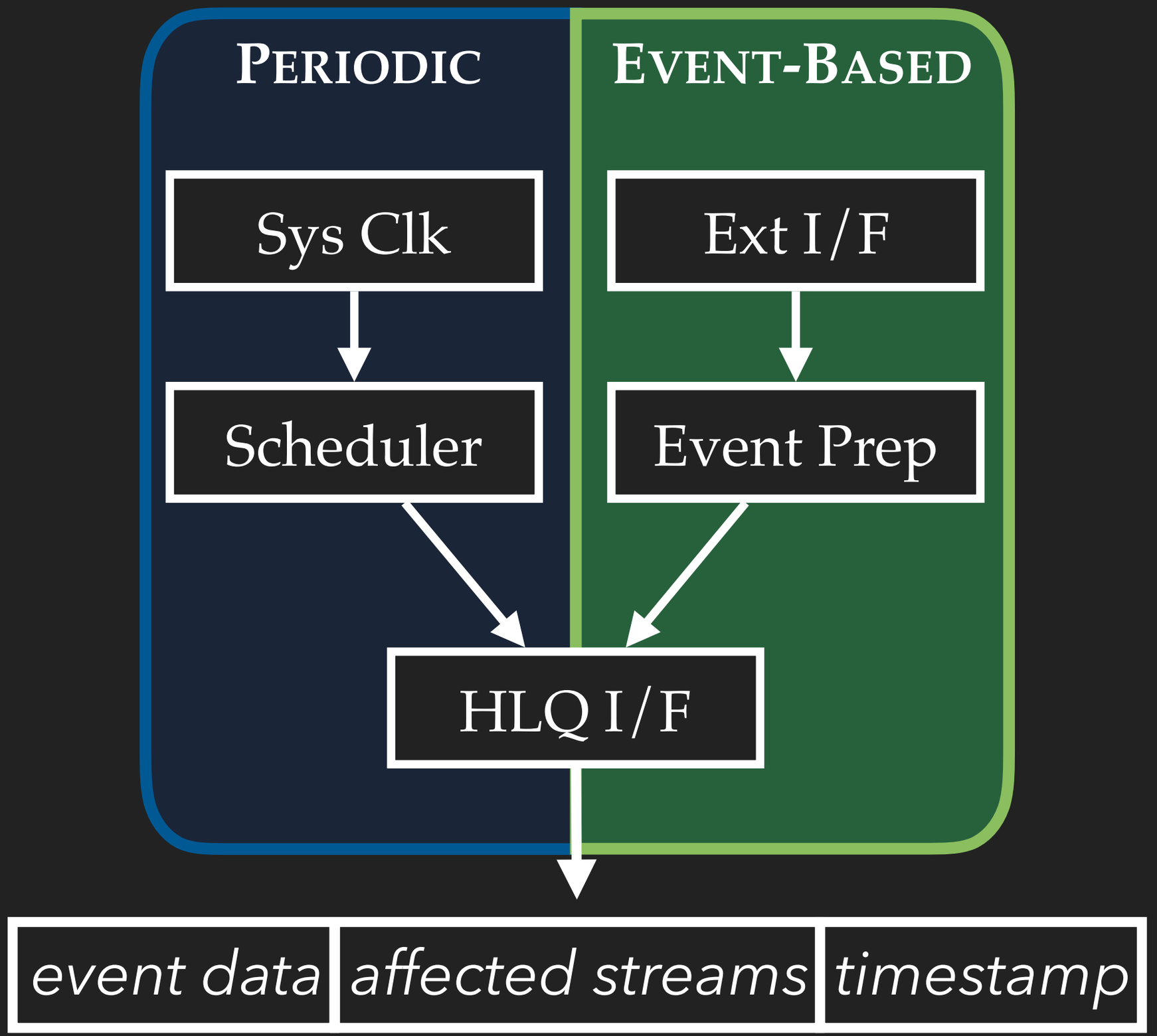
3.4,T,#,9;
110101…
4:15pm

High-Level Controller



High-Level Controller

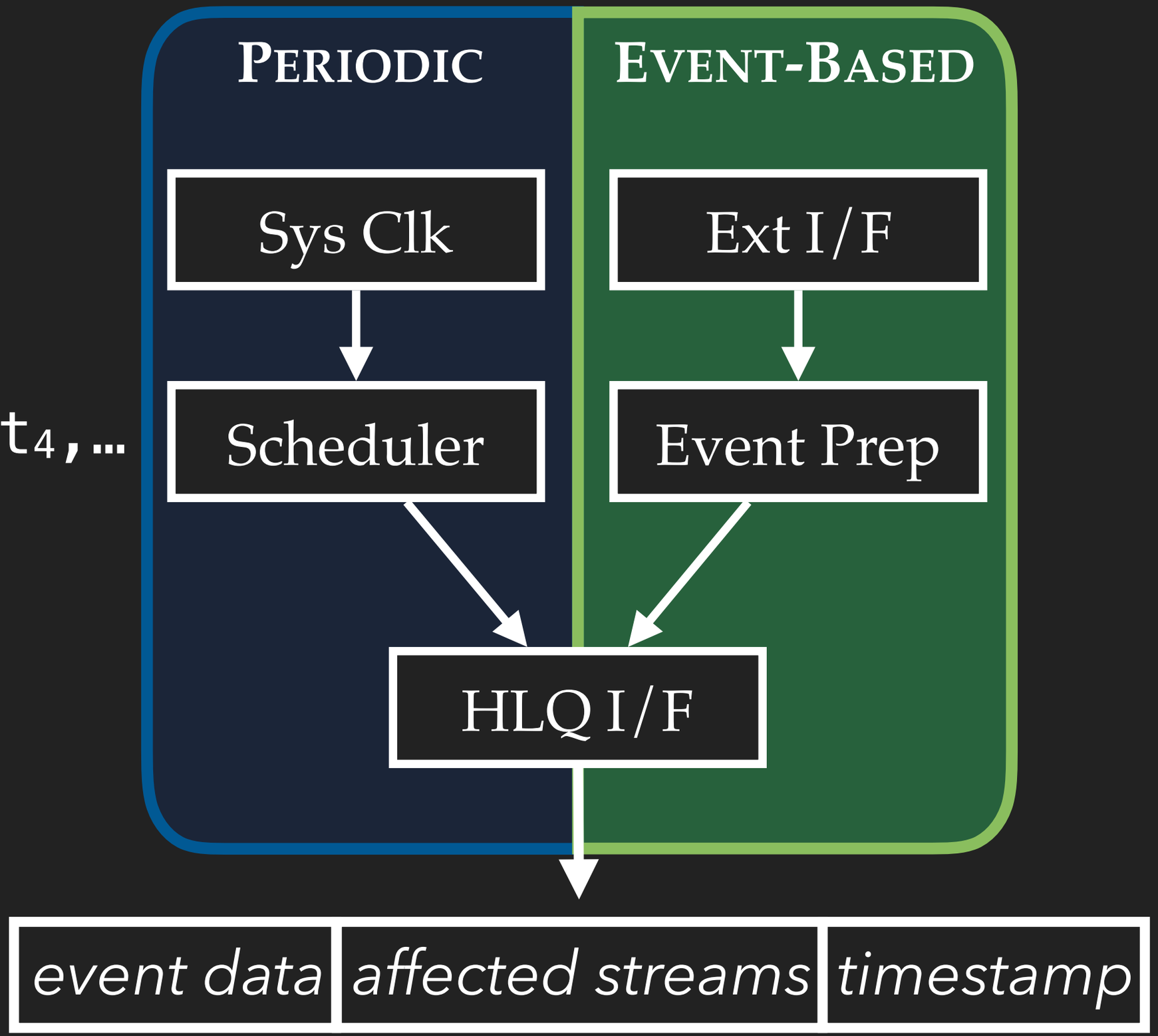
4:21pm



High-Level Controller

4:21pm

$out_1, out_3, out_4, \ldots$

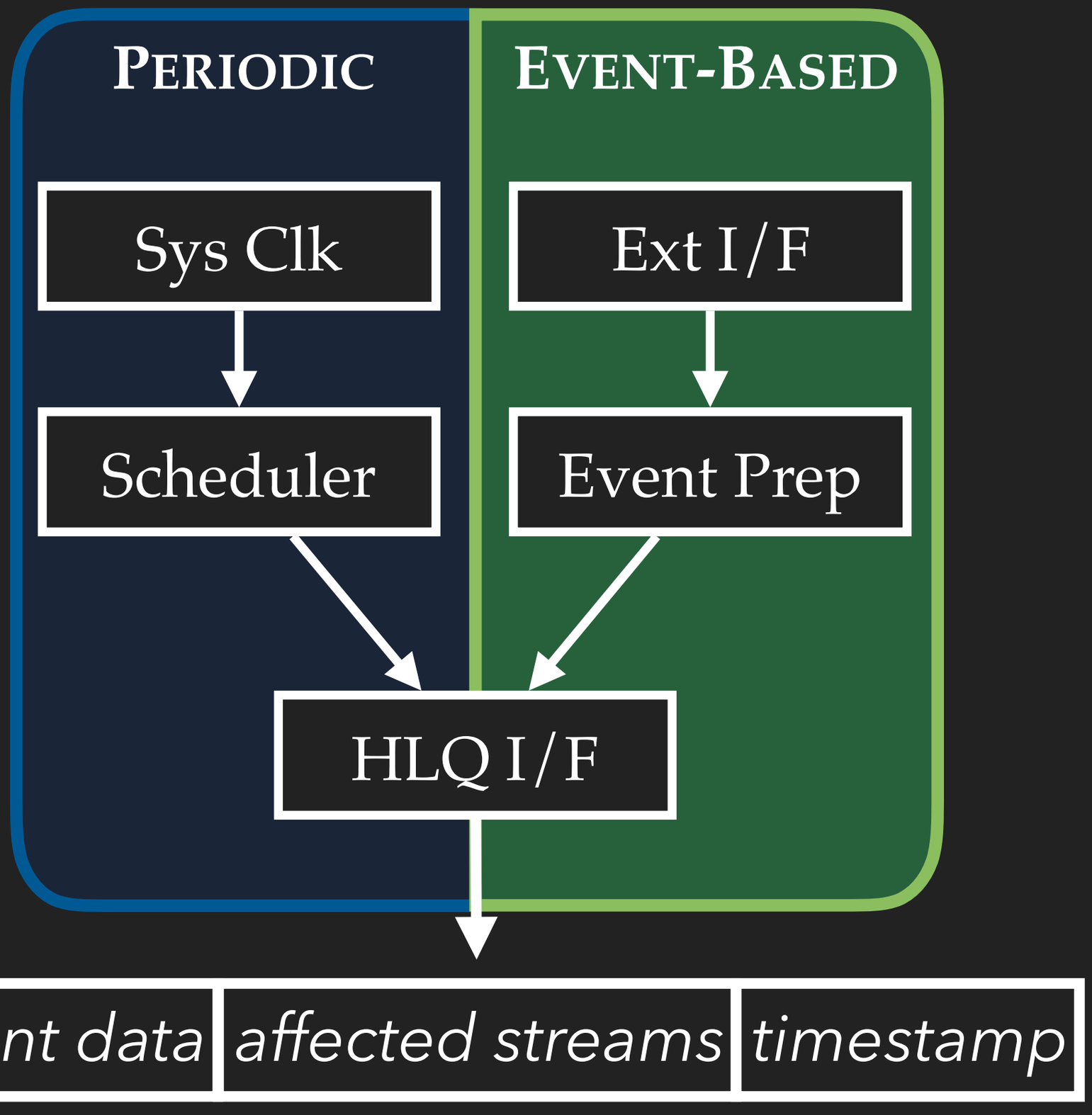


High-Level Controller

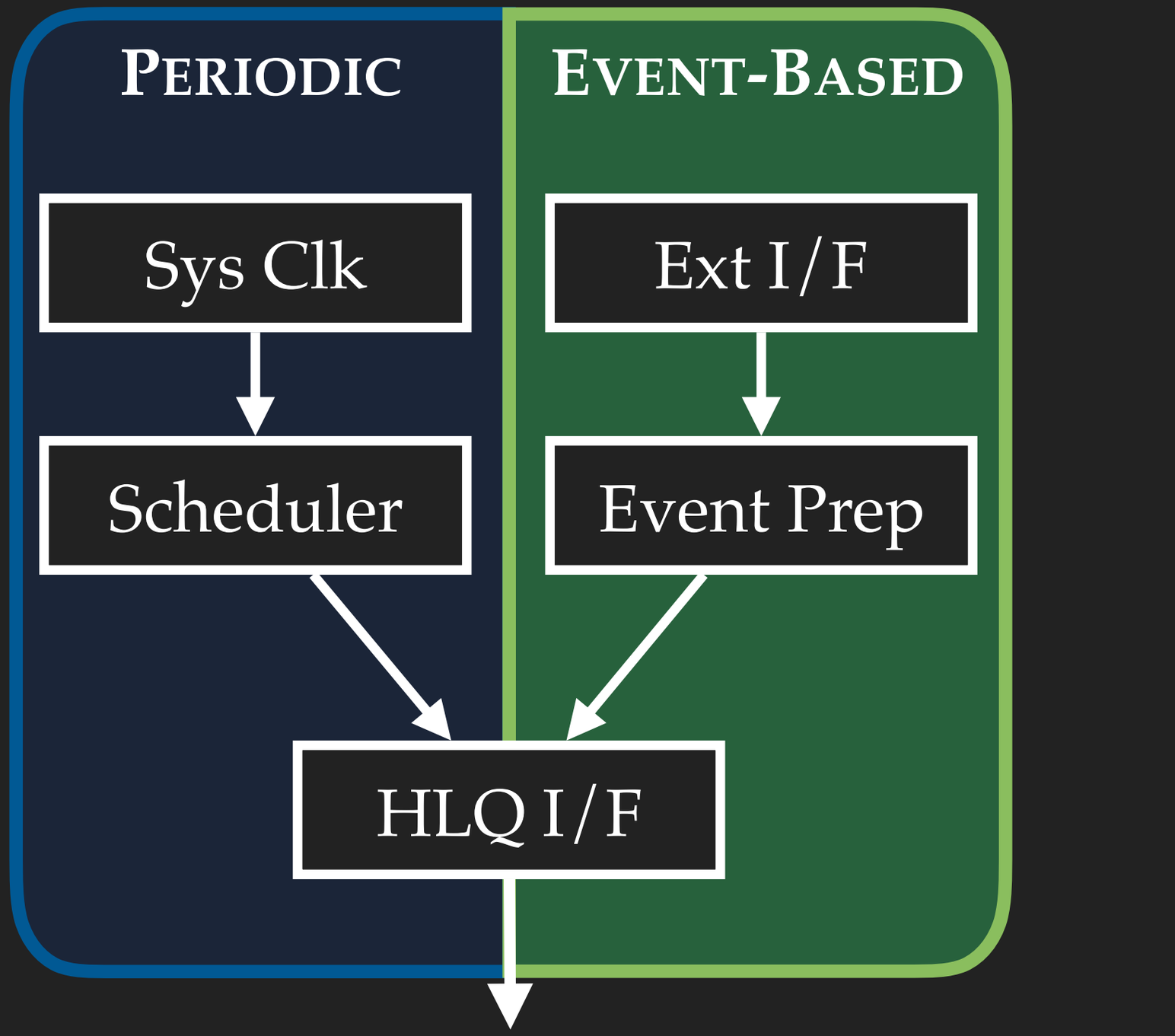
4:21pm

$out_1, out_3, out_4, \ldots$

0…0;
00001011…
4:21pm



High-Level Controller



High-Level Controller

Periodic

Sys Clk

Scheduler

Event-Based

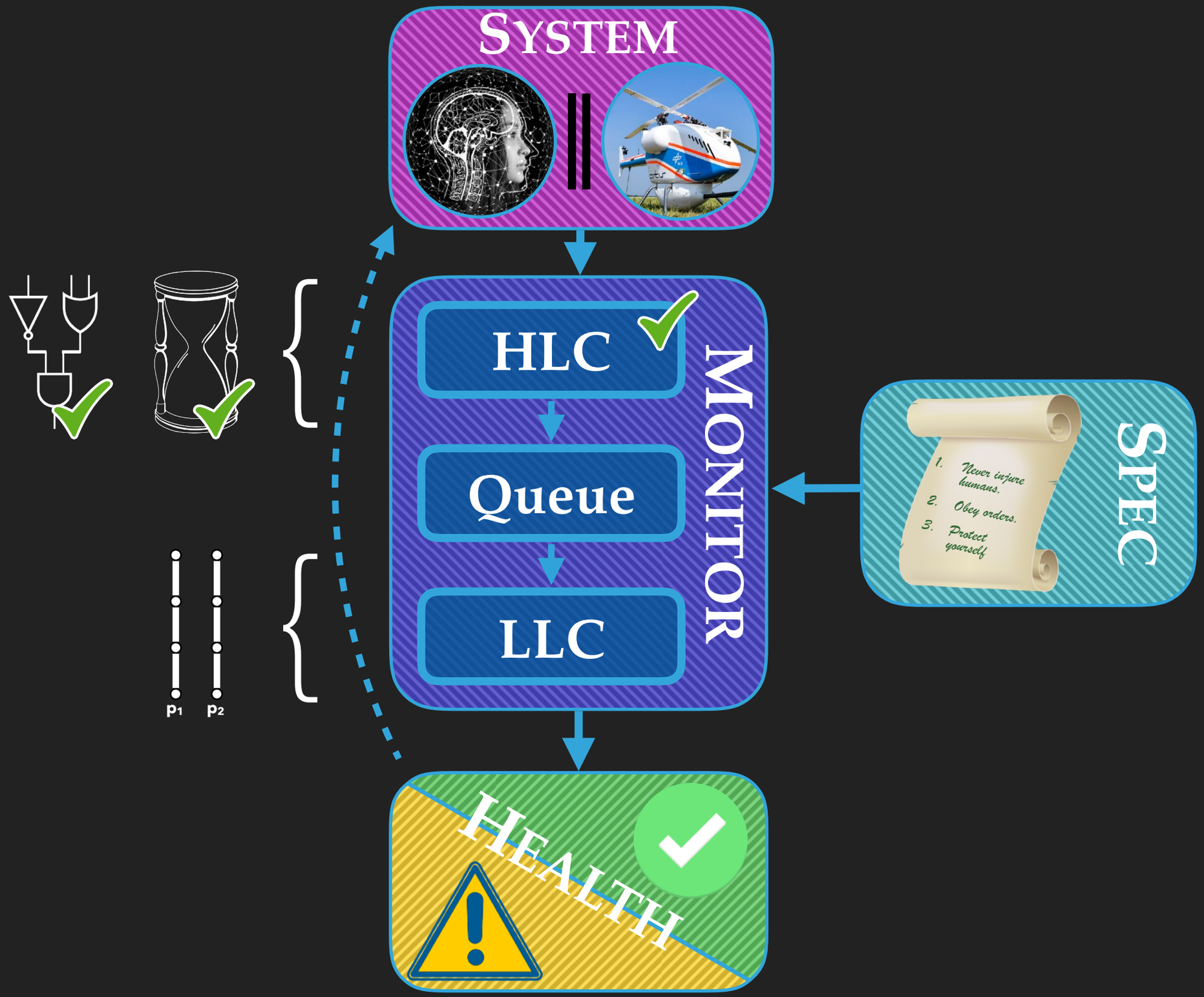
Ext I/F

Event Prep

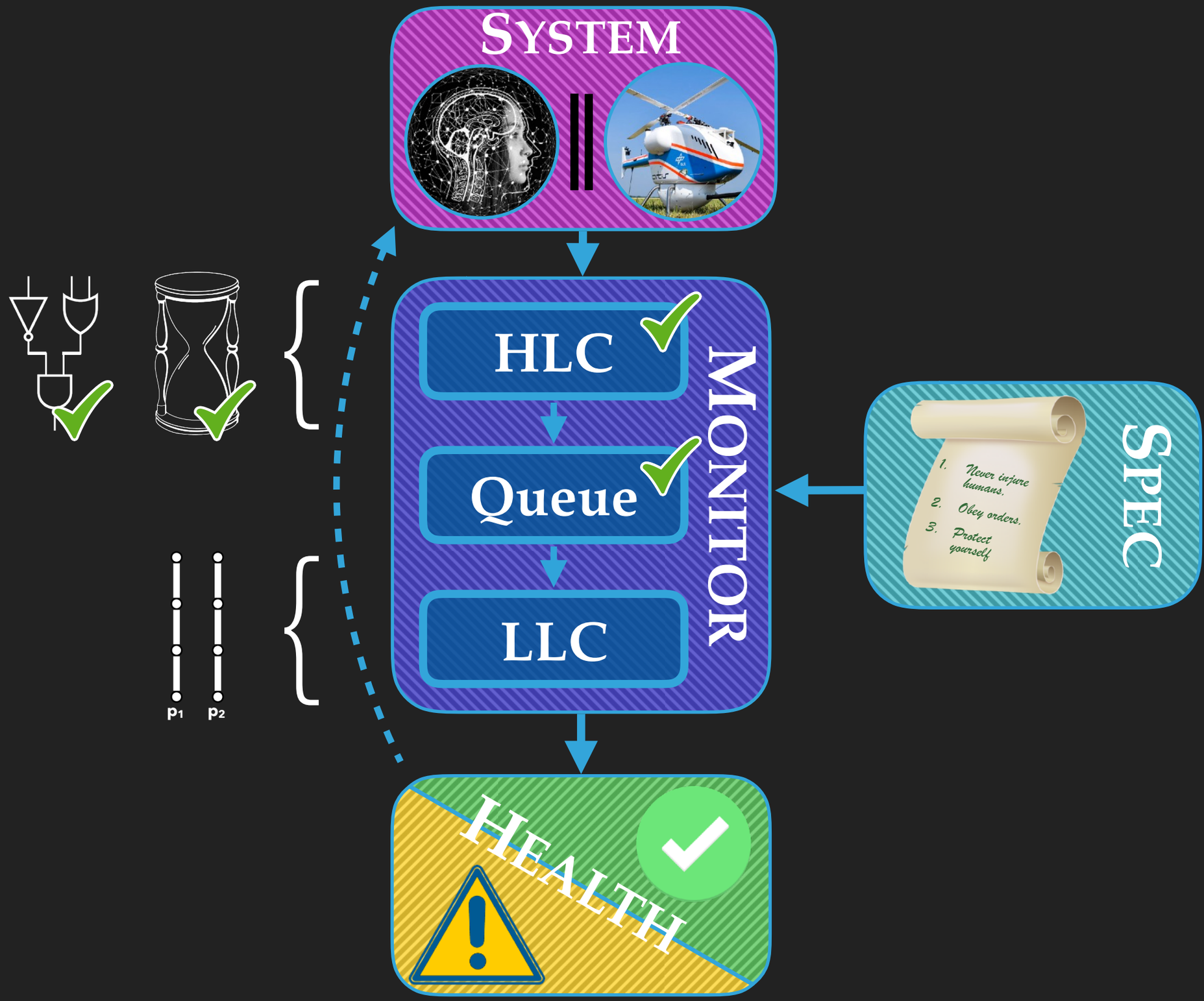
HLQ I/F

event data	*affected streams*	*timestamp*

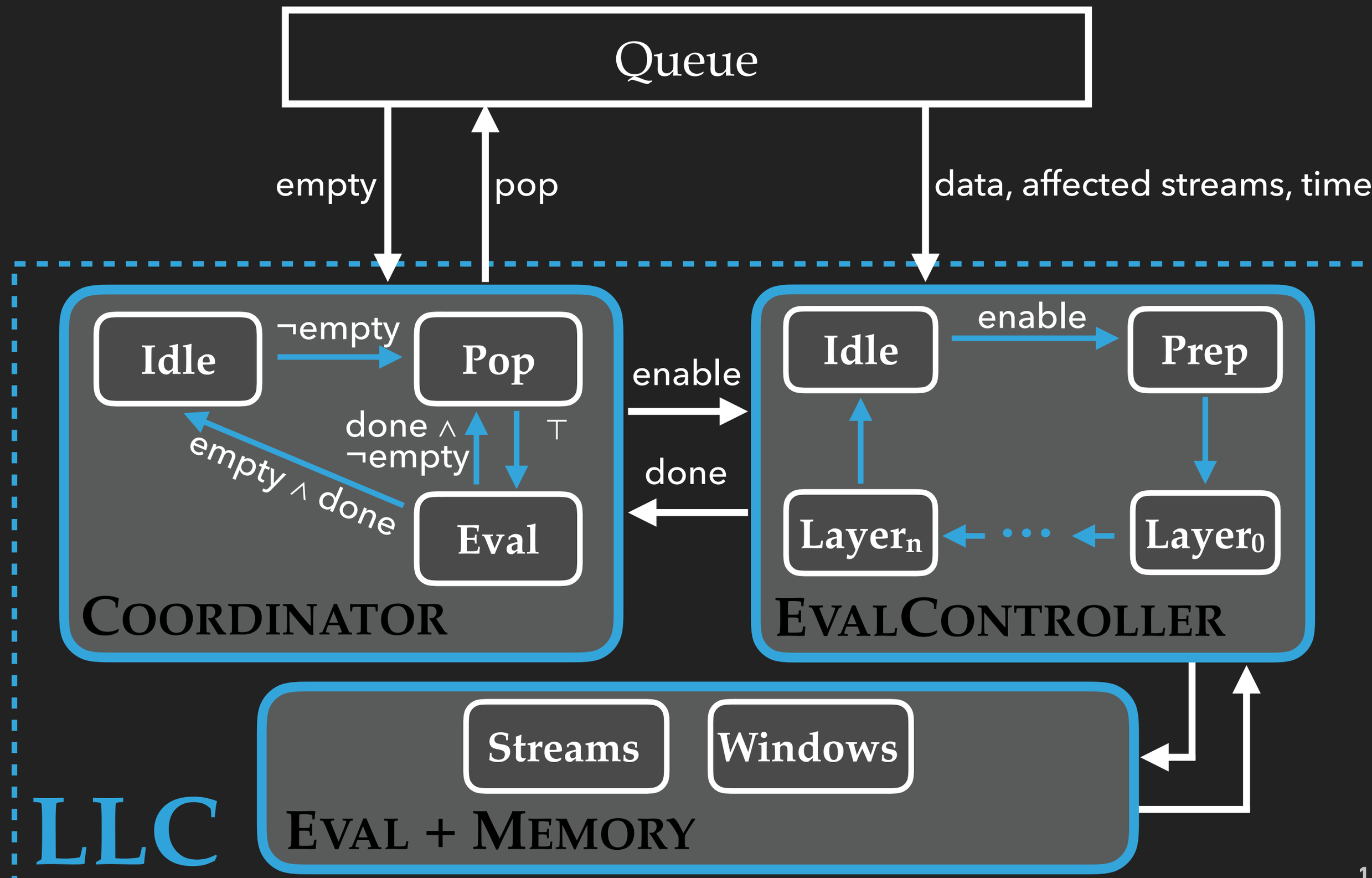
From RTLola to VHDL



From RTLola to VHDL



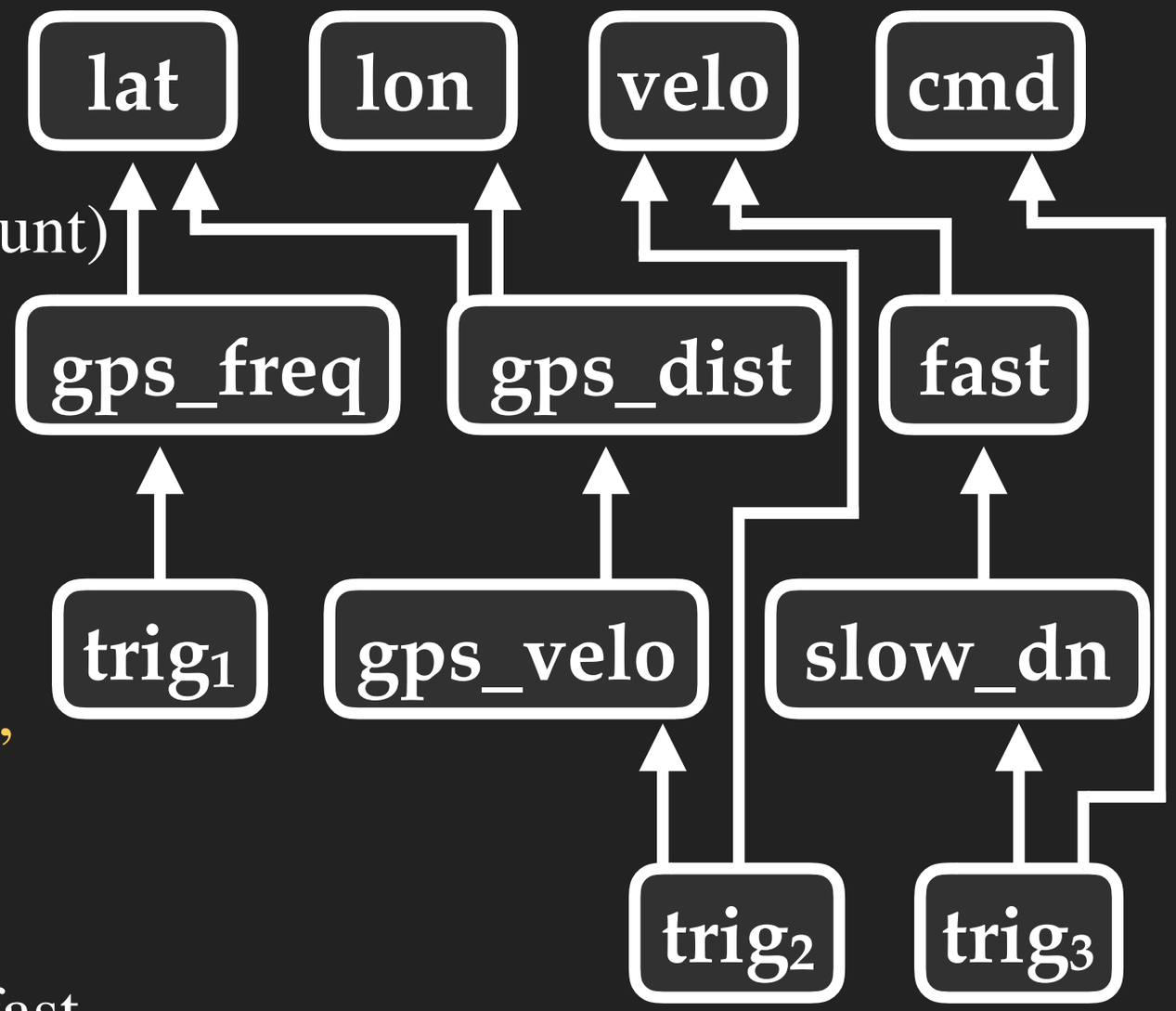
Low-Level Controller



Specification

```
input lat, lon, velo: Float64
input slow_down_cmd: Bool
output gps_freq @1Hz :=
  lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5
  “GPS frequency less than 5 Hz.”
output gps_dist := sqrt(δ(lon)² + δ(lat)²)
output gps_velo := ∇(gps_dist)
trigger abs(gps_velo - velo) > 0.1
  “Conflicting measurements for velocity.”
output fast := velo > 700
output slow_down :=
  fast.offset(by: -1).defaults(to: false) ∧ ¬fast
trigger @1Hz ¬slow_down_cmd.aggregate(over: 5s, using: ∃)
             ∧ slow_down.hold().defaults(to: false) “Spurious Slow-Down."
```

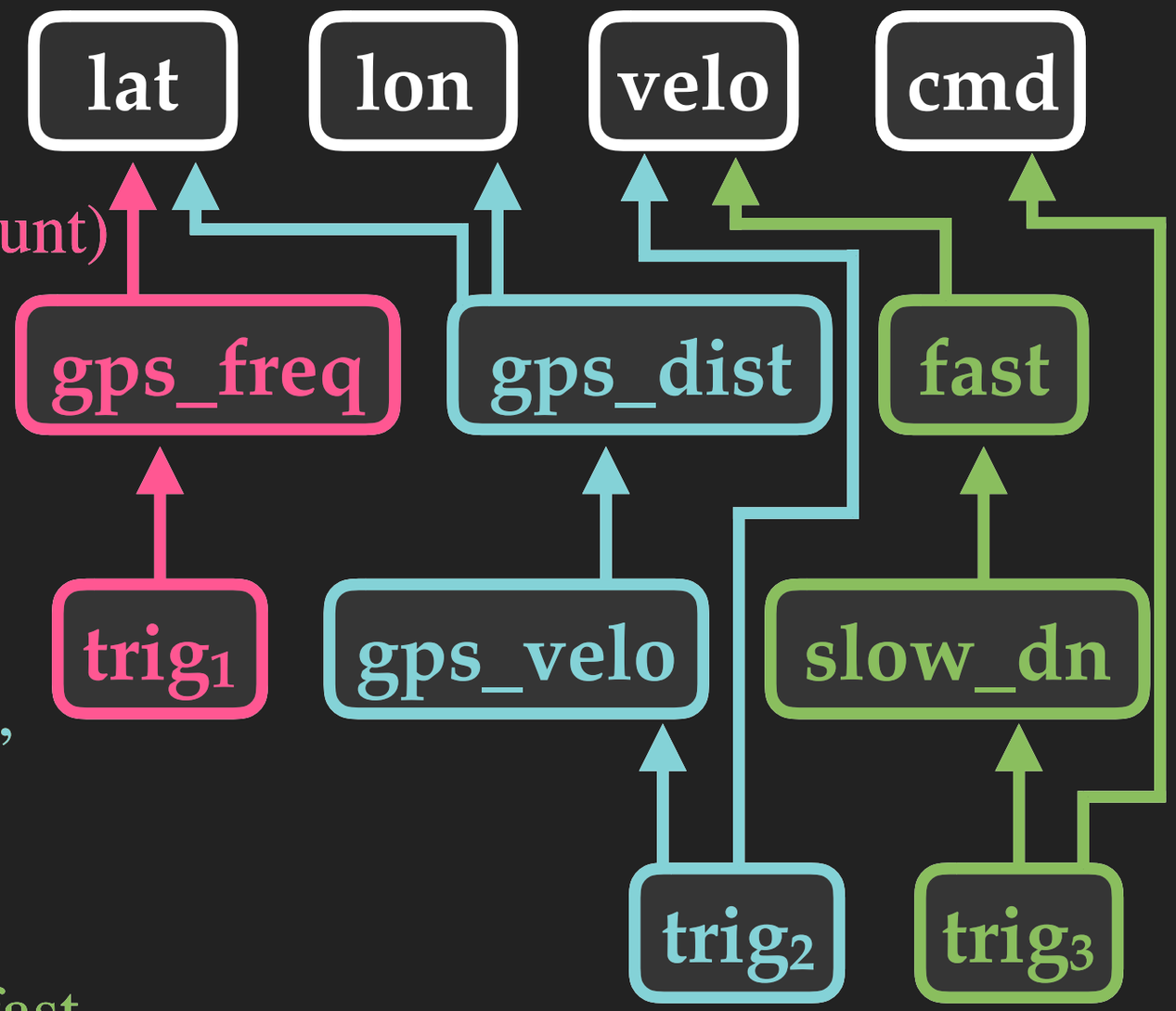
Dependency Graph



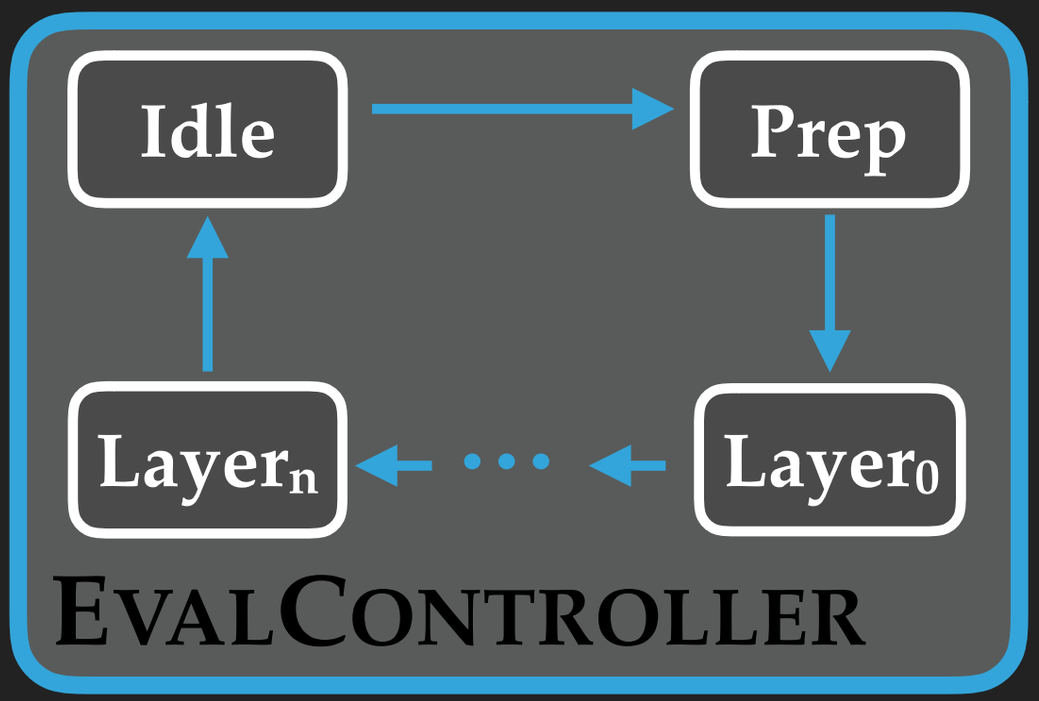
Specification

input lat, lon, velo: Float64
input slow_down_cmd: Bool
output gps_freq @1Hz :=
  lat.aggregate(over_exactly: 1s, using: count)
trigger gps_freq < 5
  “GPS frequency less than 5 Hz.”
output gps_dist := sqrt(δ(lon)² + δ(lat)²)
output gps_velo := ∇(gps_dist)
trigger abs(gps_velo - velo) > 0.1
  “Conflicting measurements for velocity.”
output fast := velo > 700
output slow_down :=
  fast.offset(by: -1).defaults(to: false) ∧ ¬fast
trigger @1Hz ¬slow_down_cmd.aggregate(over: 5s, using: ∃)
  ∧ slow_down.hold().defaults(to: false) “Spurious Slow-Down.”

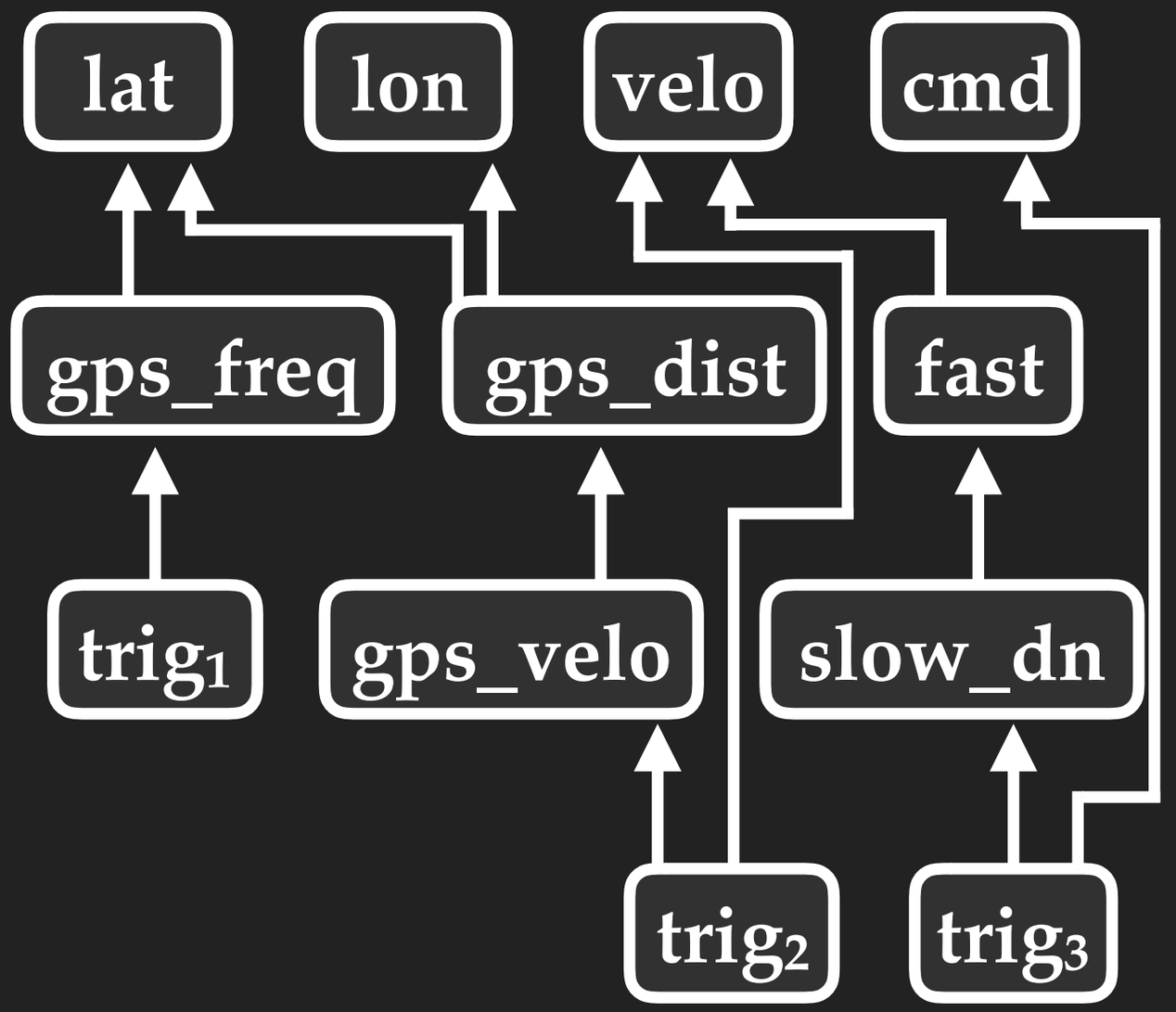
Dependency Graph



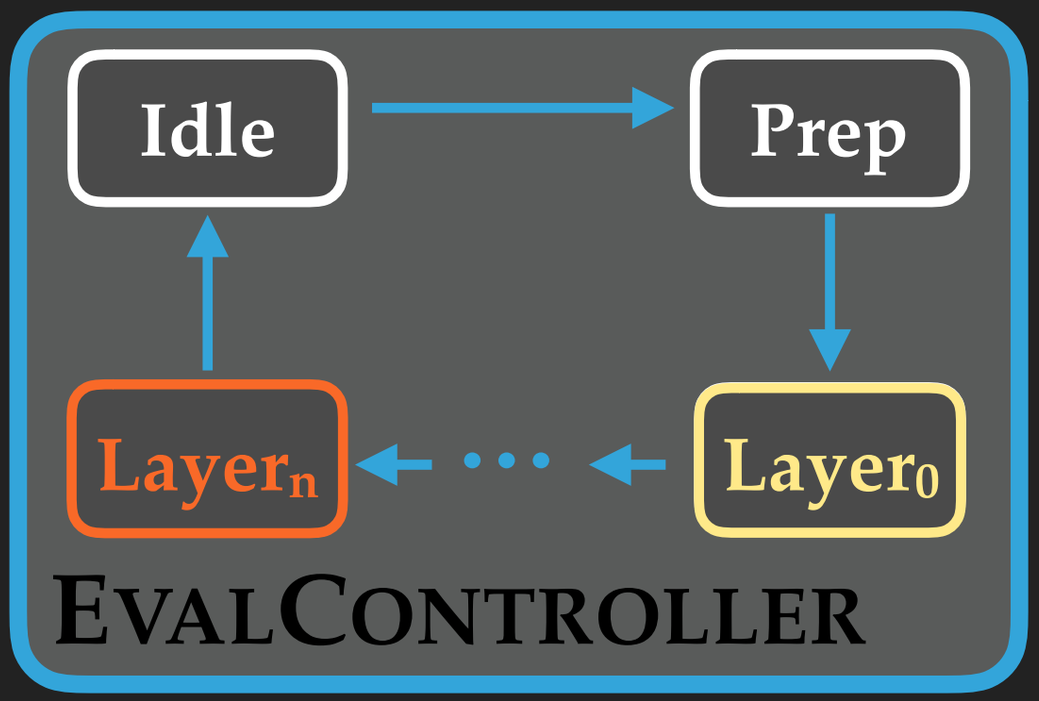
Low-Level Controller



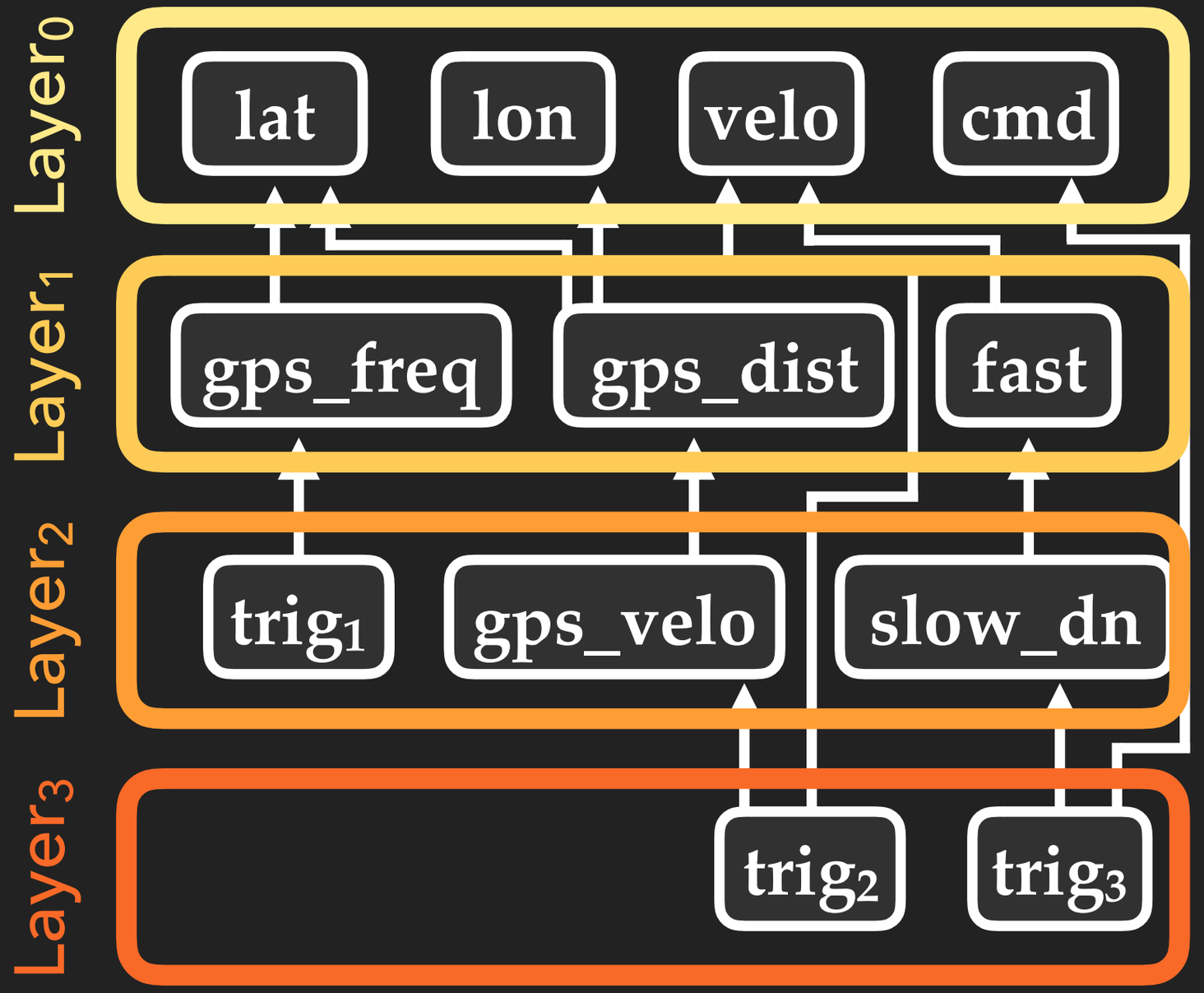
Dependency Graph



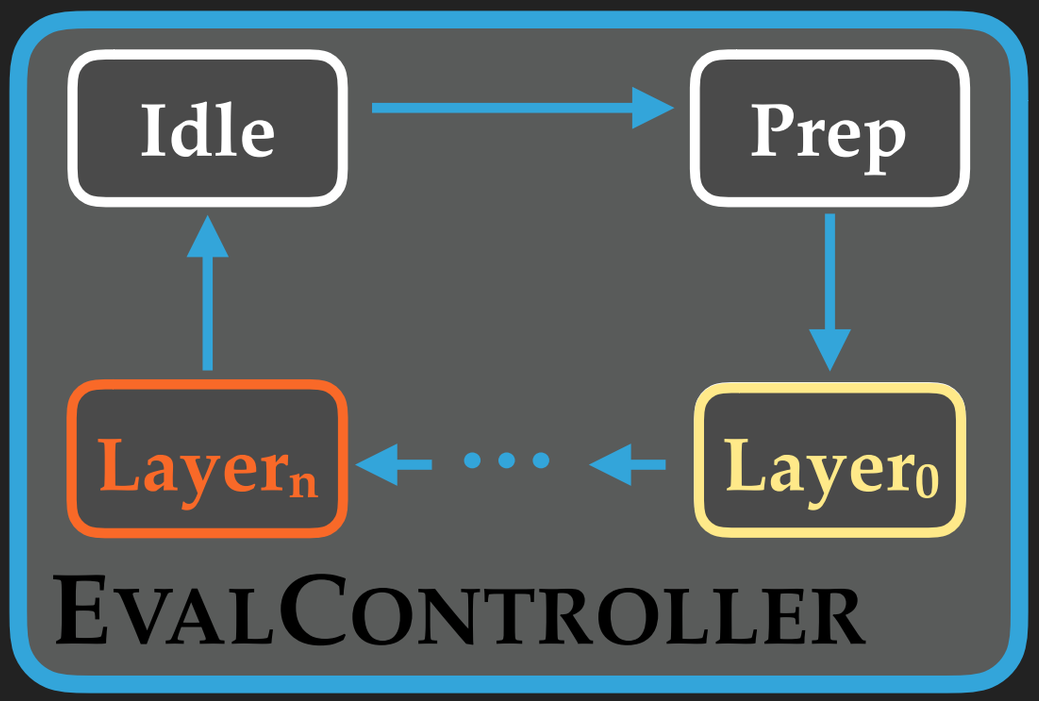
Low-Level Controller

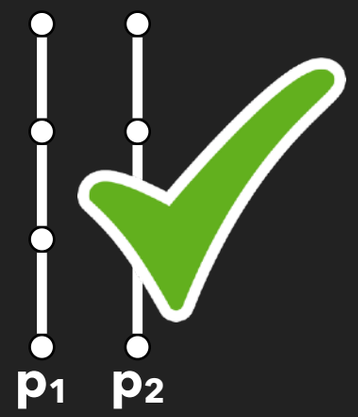


Dependency Graph

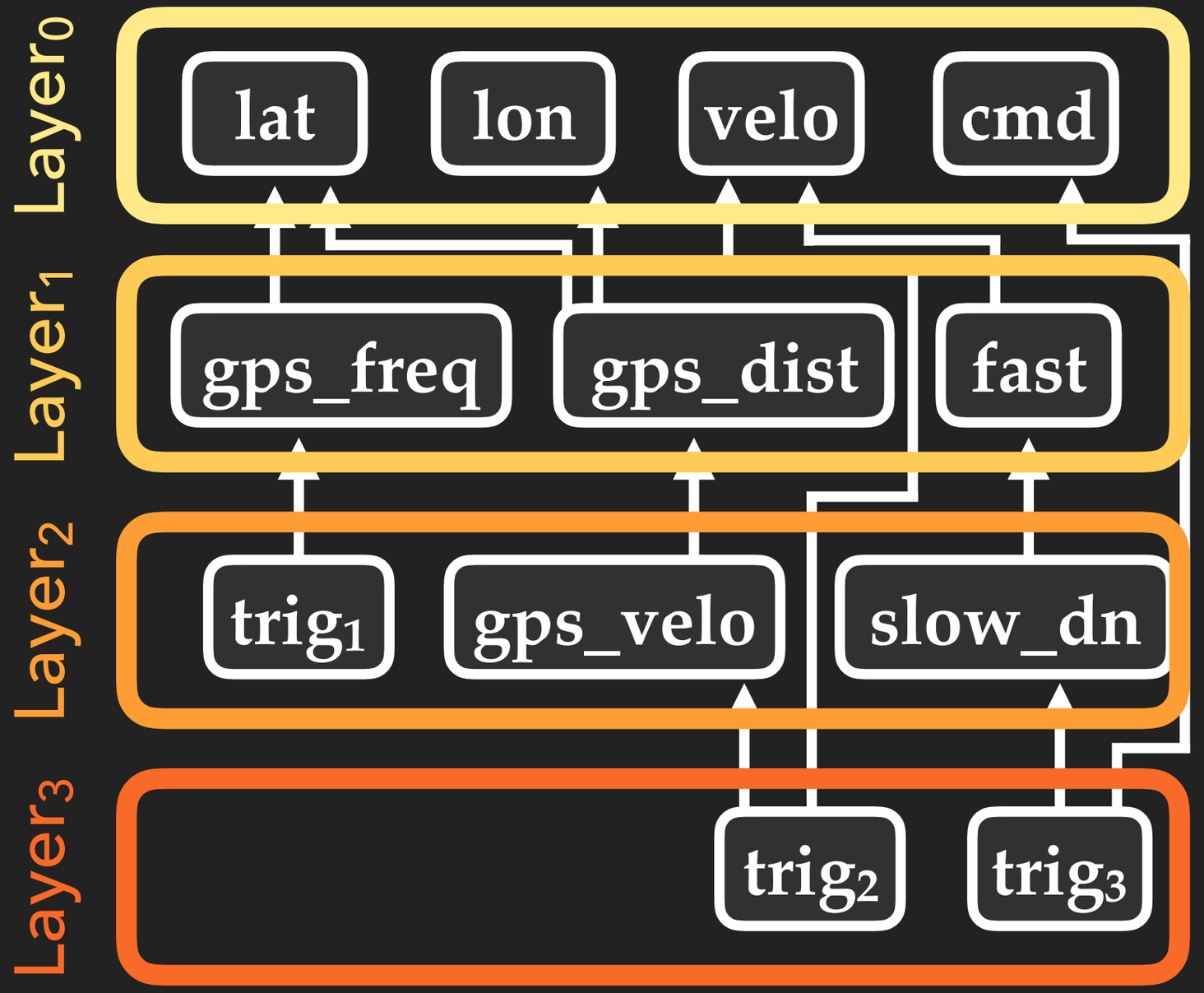


Low-Level Controller

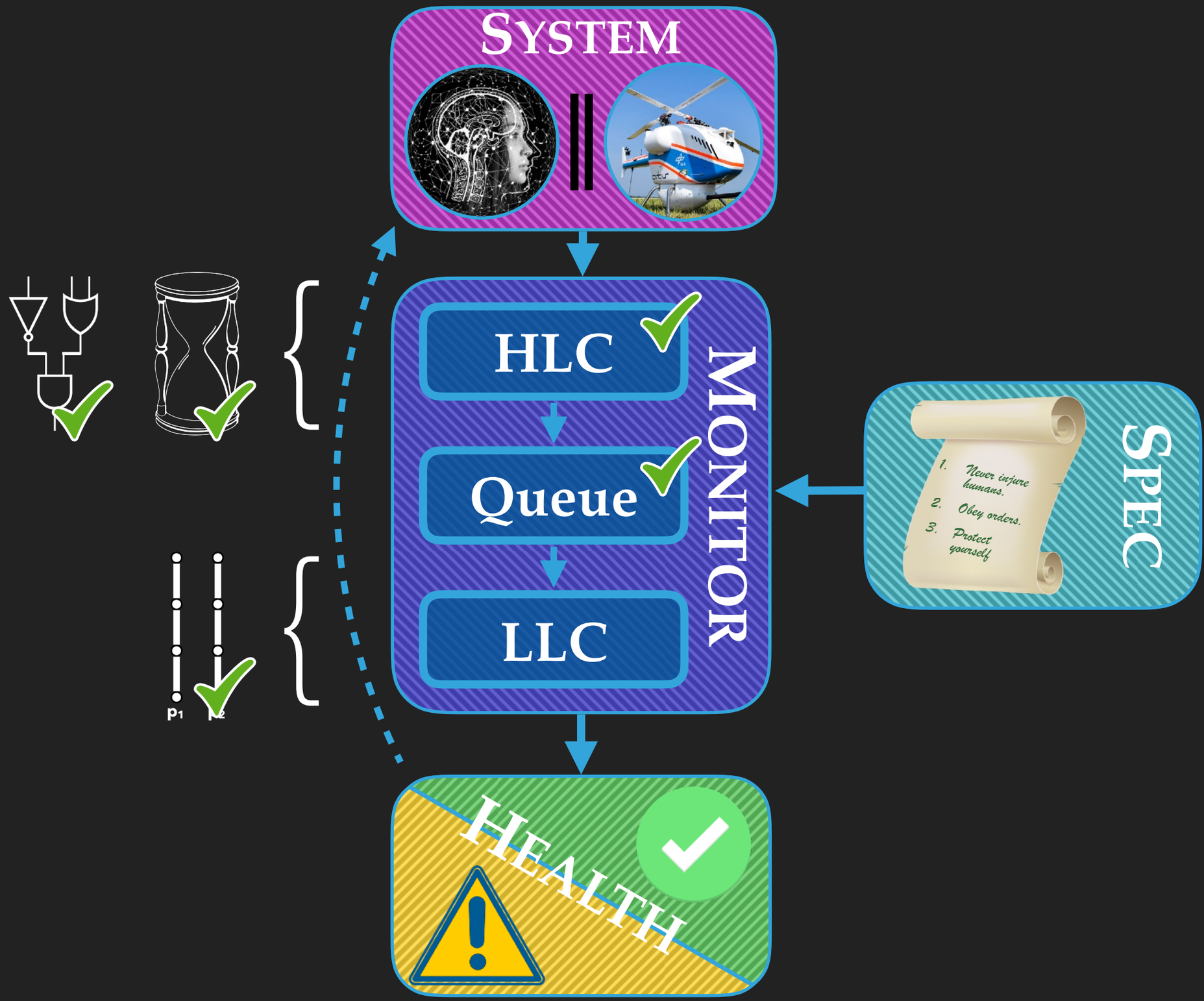




Dependency Graph

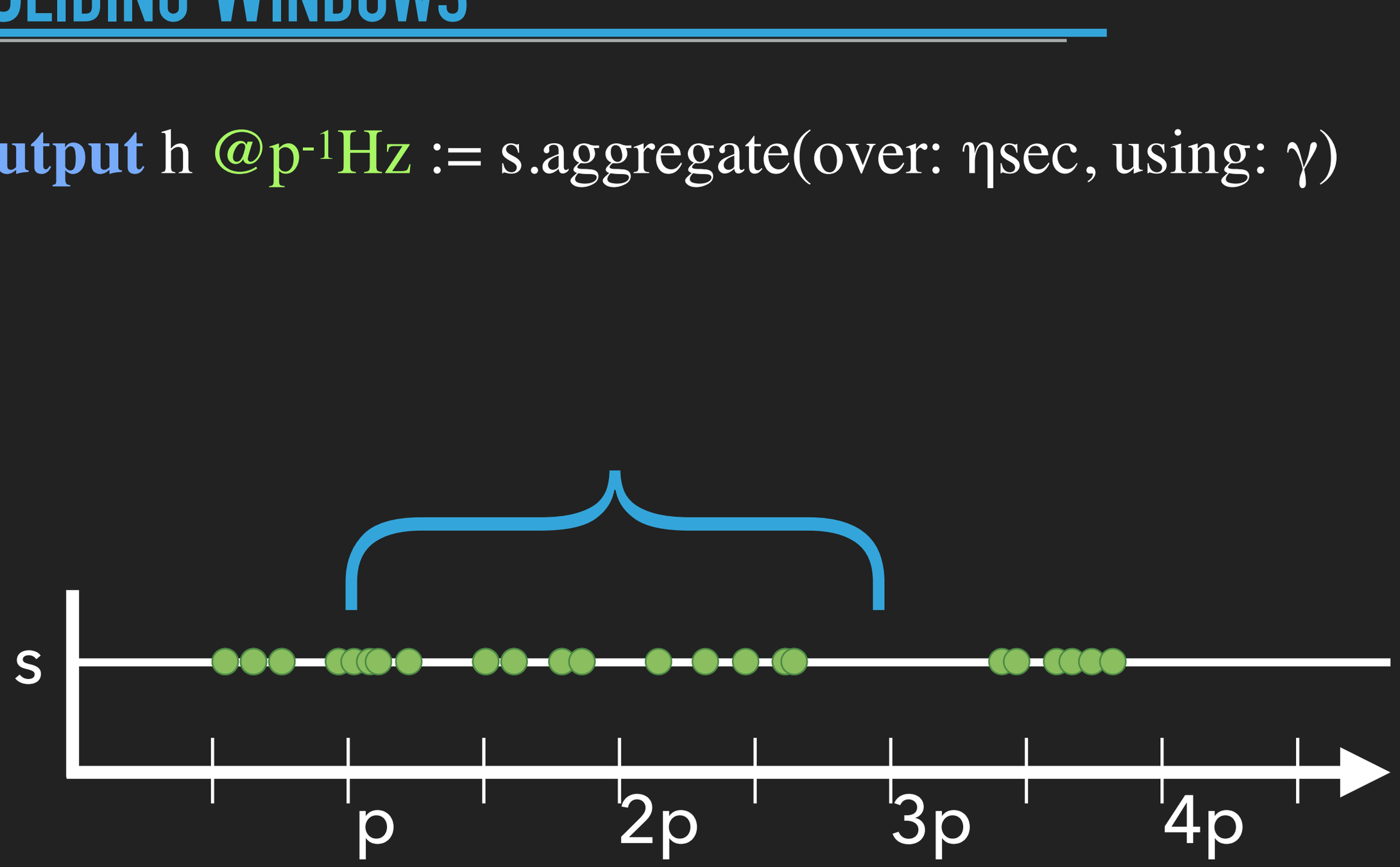


From RTLola to VHDL



Sliding Windows

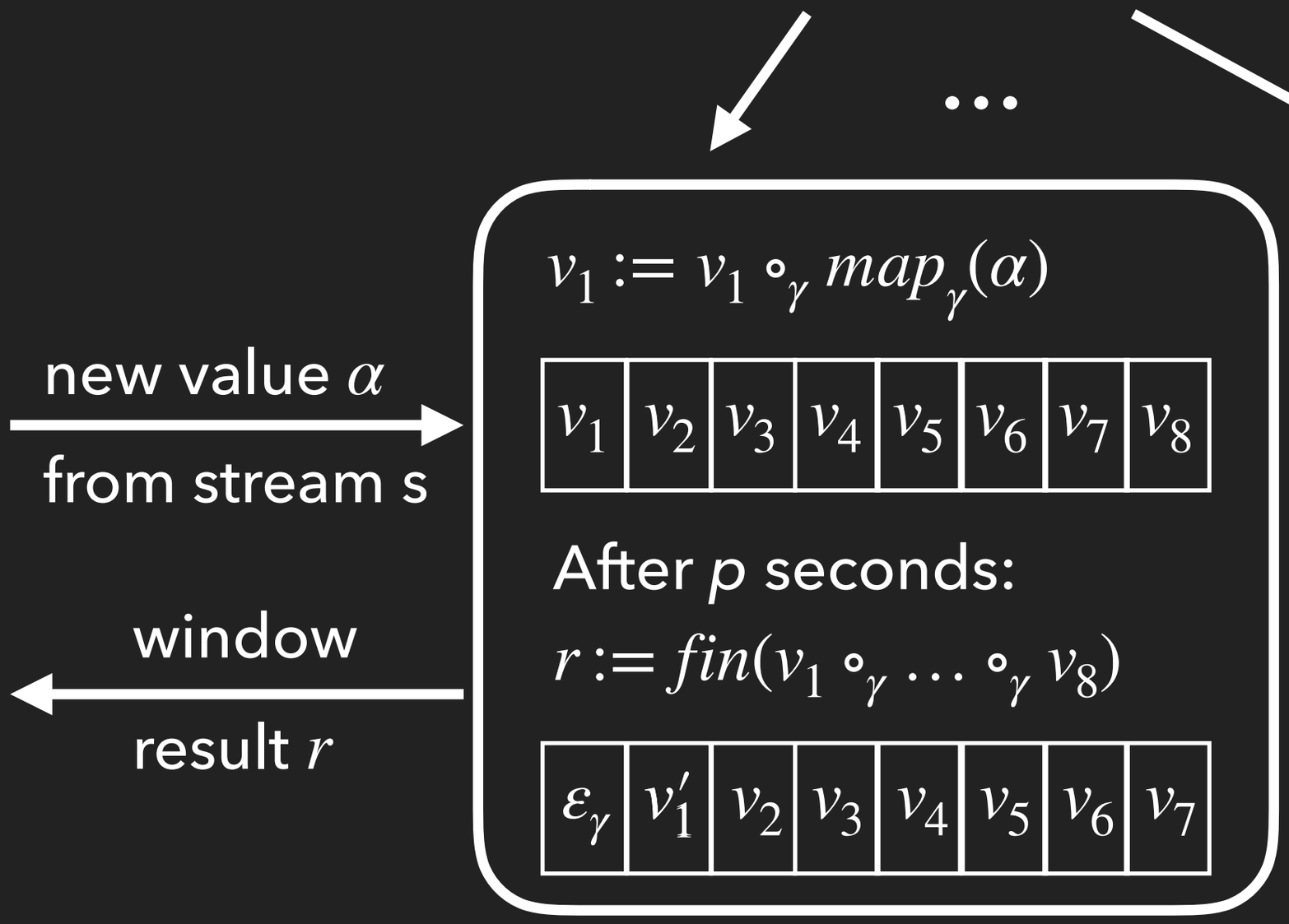
output h @p^{-1}Hz := s.aggregate(over: ηsec, using: γ)



- Li et al.: “*No Pane, No Gain: Efficient Evaluation of Sliding-window Aggregates over Data Streams”*, SIGMOD Rec. 2005
- Meertens: “*Algorithmics: Towards programming as a mathematical activity”,* 1986
- Schwenger: “*Let’s not Trust Experience Blindly: Formal Monitoring of Humans and other CPS*”, Master Thesis 2019
- Faymonville, Finkbeiner, Schledjewski, Schwenger, Stenger, Tentrup, Torfah, “*StreamLAB: Stream-based Monitoring of Cyber-physical Systems*”, CAV 2019

Honorable Mention: Sliding Windows

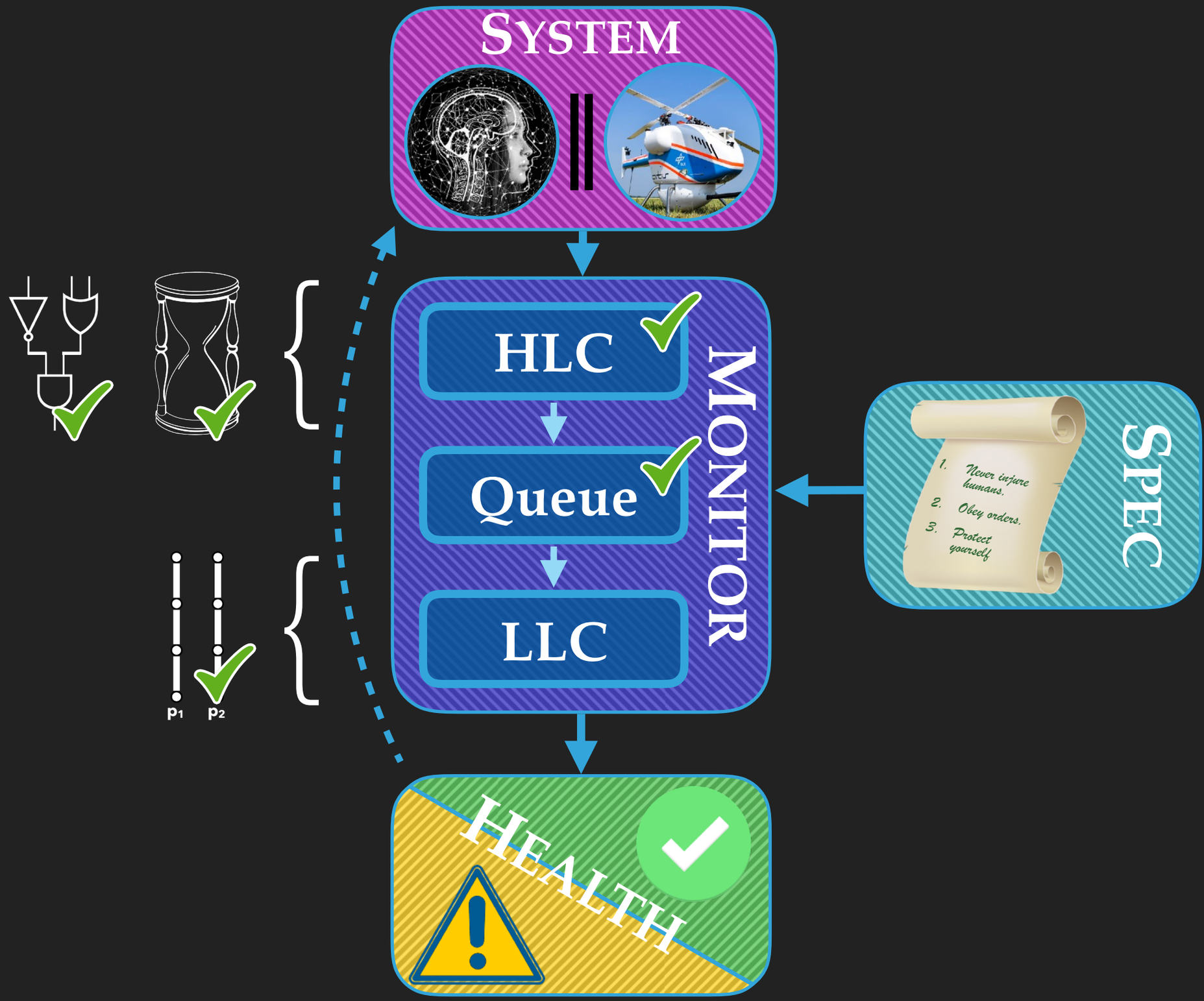
output h @p^{-1}Hz := s.aggregate(over: ηsec, using: γ)



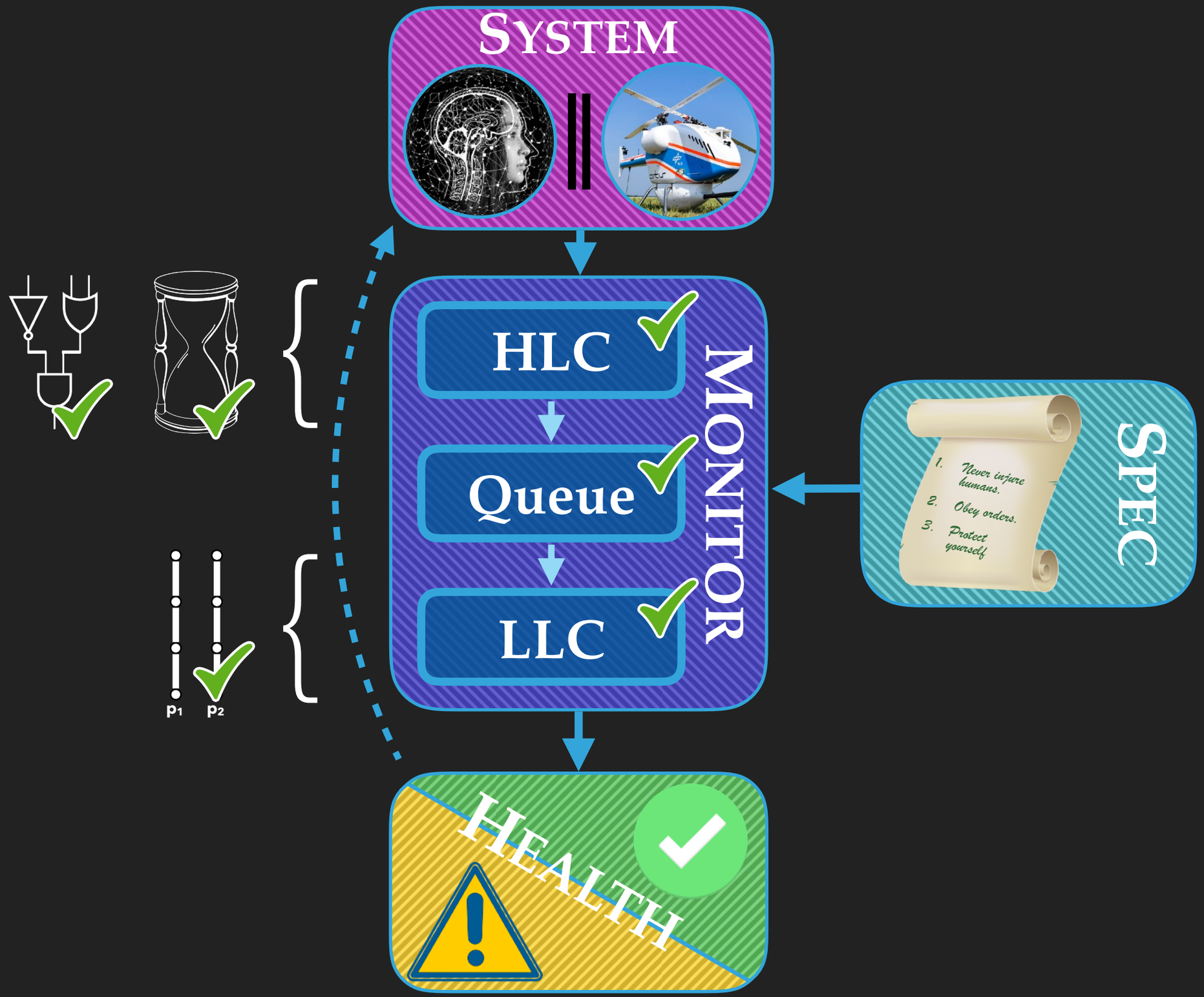
...

- Li et al.: "*No Pane, No Gain: Efficient Evaluation of Sliding-window Aggregates over Data Streams"*, SIGMOD Rec. 2005
- Meertens: "*Algorithmics: Towards programming as a mathematical activity"*, 1986
- Schwenger: "*Let's not Trust Experience Blindly: Formal Monitoring of Humans and other CPS*", Master Thesis 2019
- Faymonville, Finkbeiner, Schledjewski, Schwenger, Stenger, Tentrup, Torfah, "*StreamLAB: Stream-based Monitoring of Cyber-physical Systems*", CAV 2019

From RTLola to VHDL



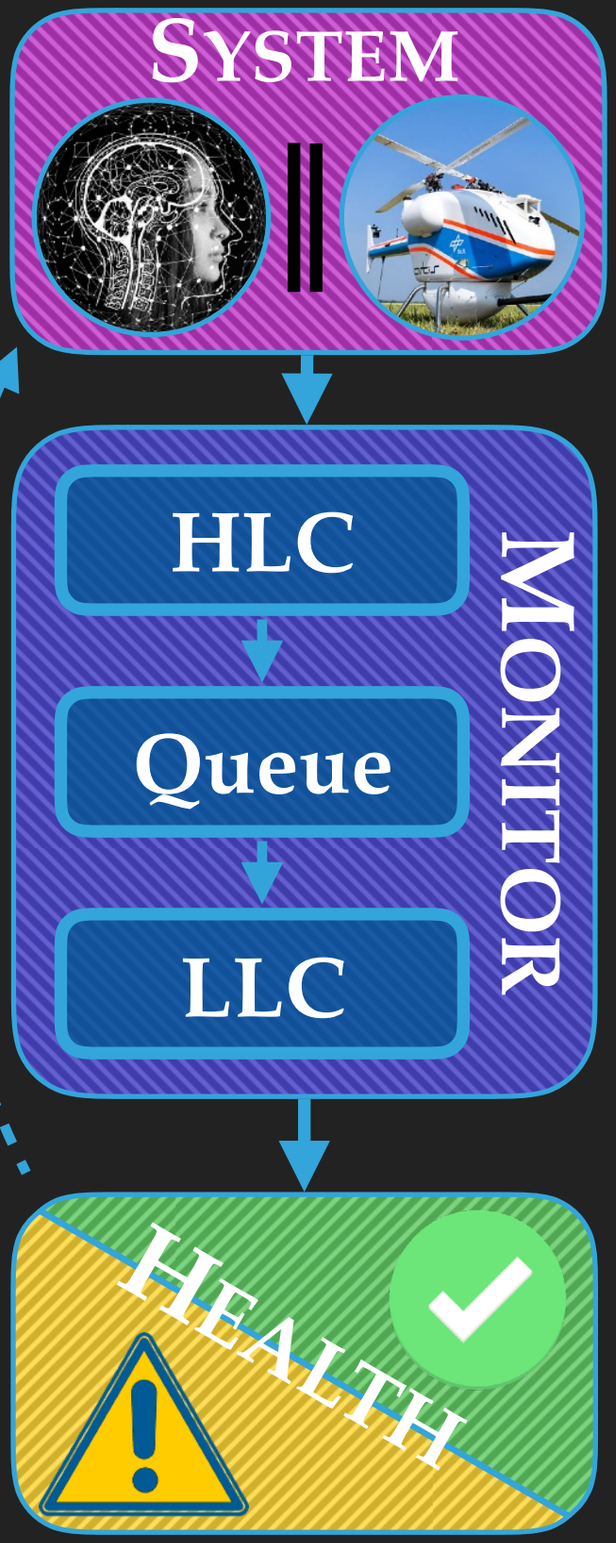
From RTLola to VHDL

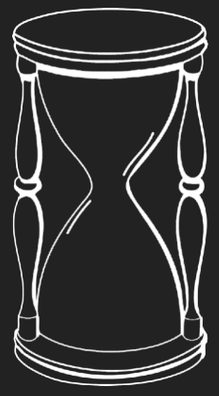


Evaluation

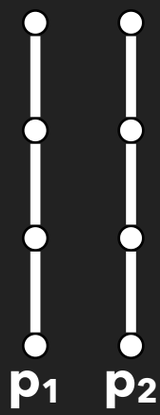
		FF	LUT	MUX	CA	MULT	Pwr [W]	Time [µs]
Drone	Mon	3036	3685	26	656	10	1.620	4.28
	HLC	901	156	0	22	0		
	Q	543	442	0	43	0		
	LLC	1281	2820	0	576	10		
Network	Mon	1905	1533	23	226	23	1.570	3.20
	HLC	550	161	0	37	0		
	Q	330	342	0	28	0		
	LLC	895	927	0	161	0		
Cmd Resp Par	Mon	6379	13794	0	849	0	1.582	3.77
	HLC	936	232	0	30	0		
	Q	540	326	0	28	0		
	LLC	4903	13236	0	971	0		
Cmd Resp Seq	Mon	6909	14768	0	851	0	1.581	43.83
	HLC	936	232	0	30	0		
	Q	534	326	0	28	0		
	LLC	5433	14210	0	973	0		

Conclusion





Periodic versus Event-Based



Utilize Parallel Execution

Reduce Circuit Cost

Outlook

LEARN MORE:
stream-lab.eu