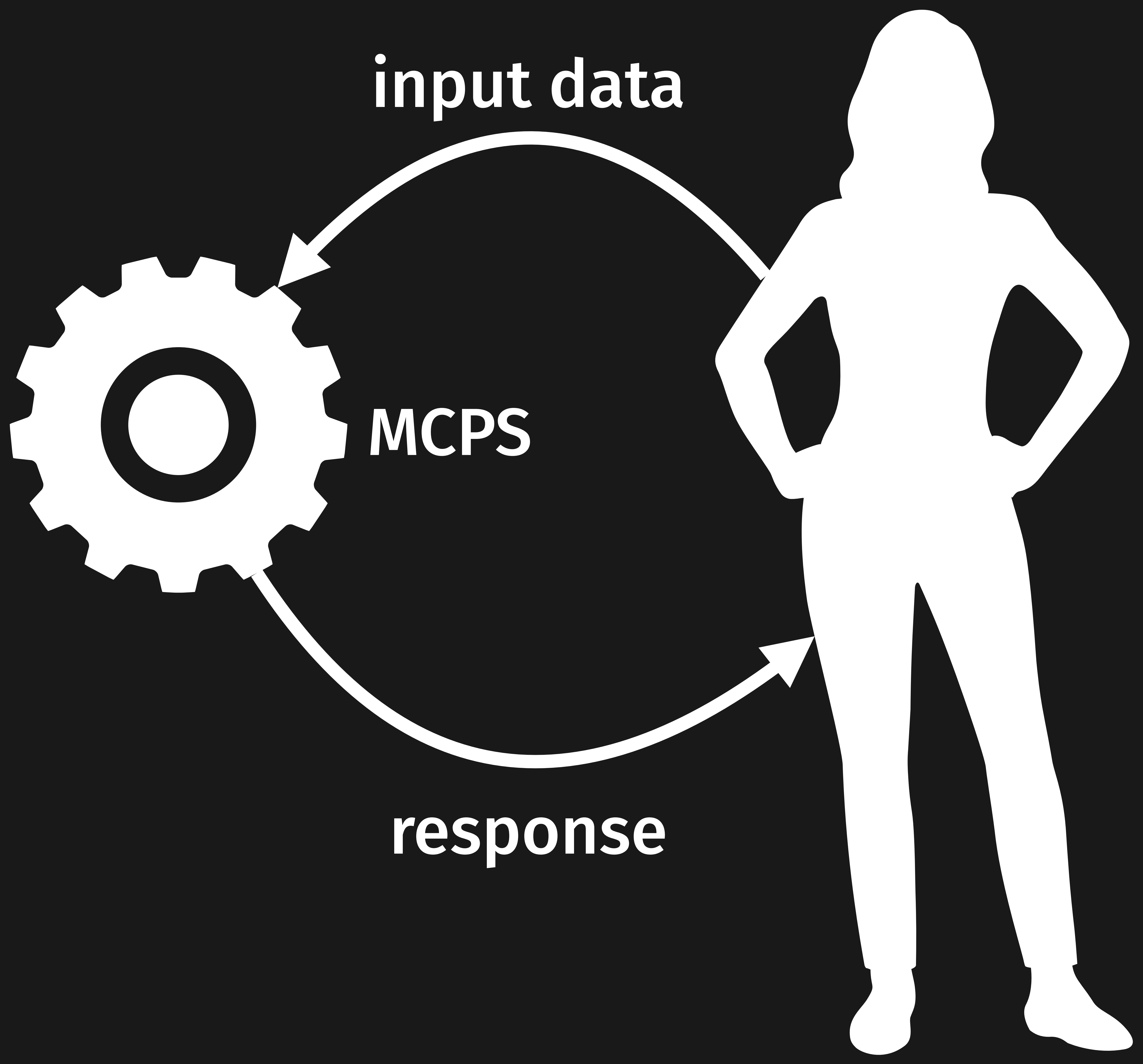
WIP

Robust monitoring for Medical CPS

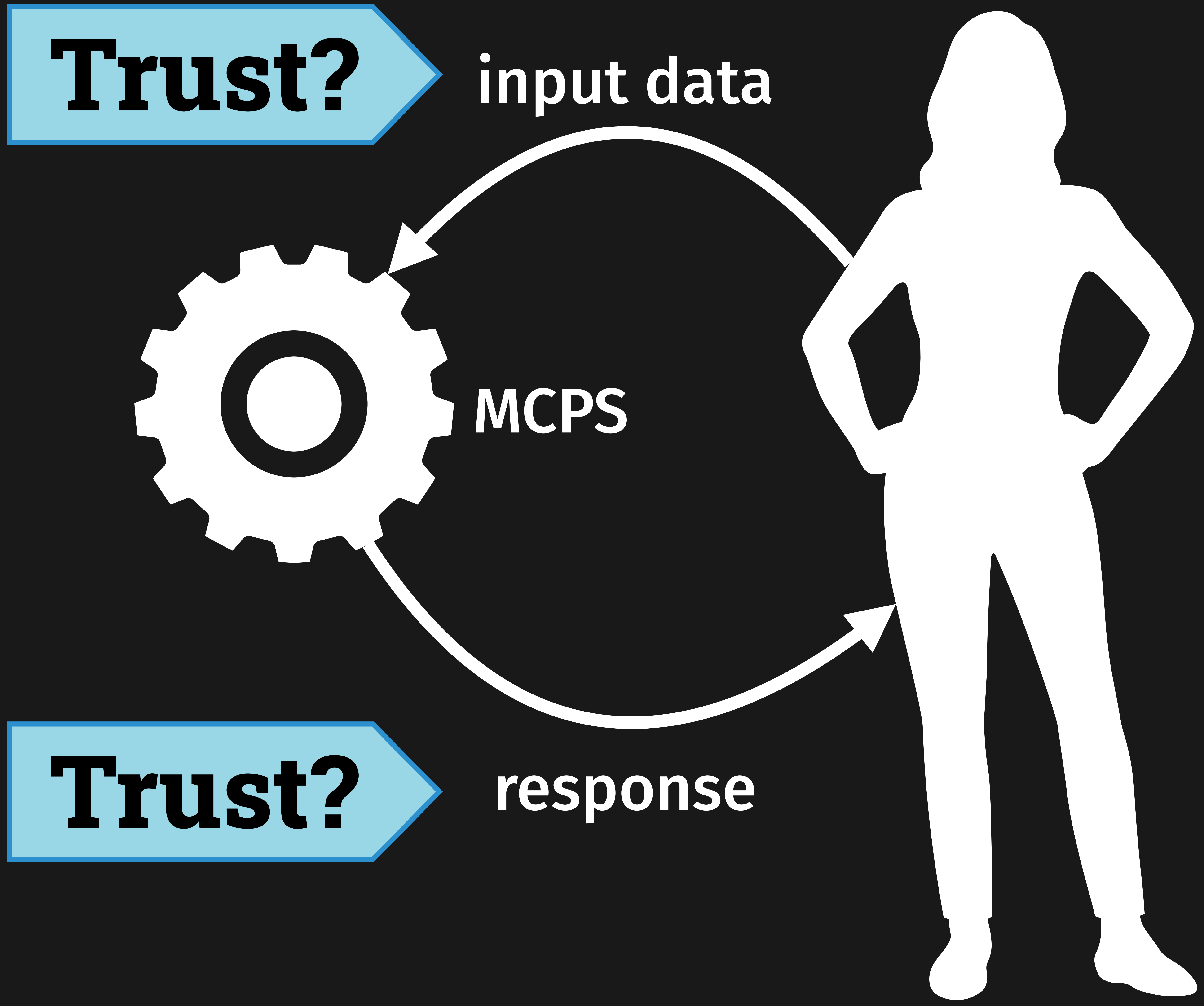
Bernd Finkbeiner, Andreas Keller, Jessica Schmidt,
Maximilian Schwenger

CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

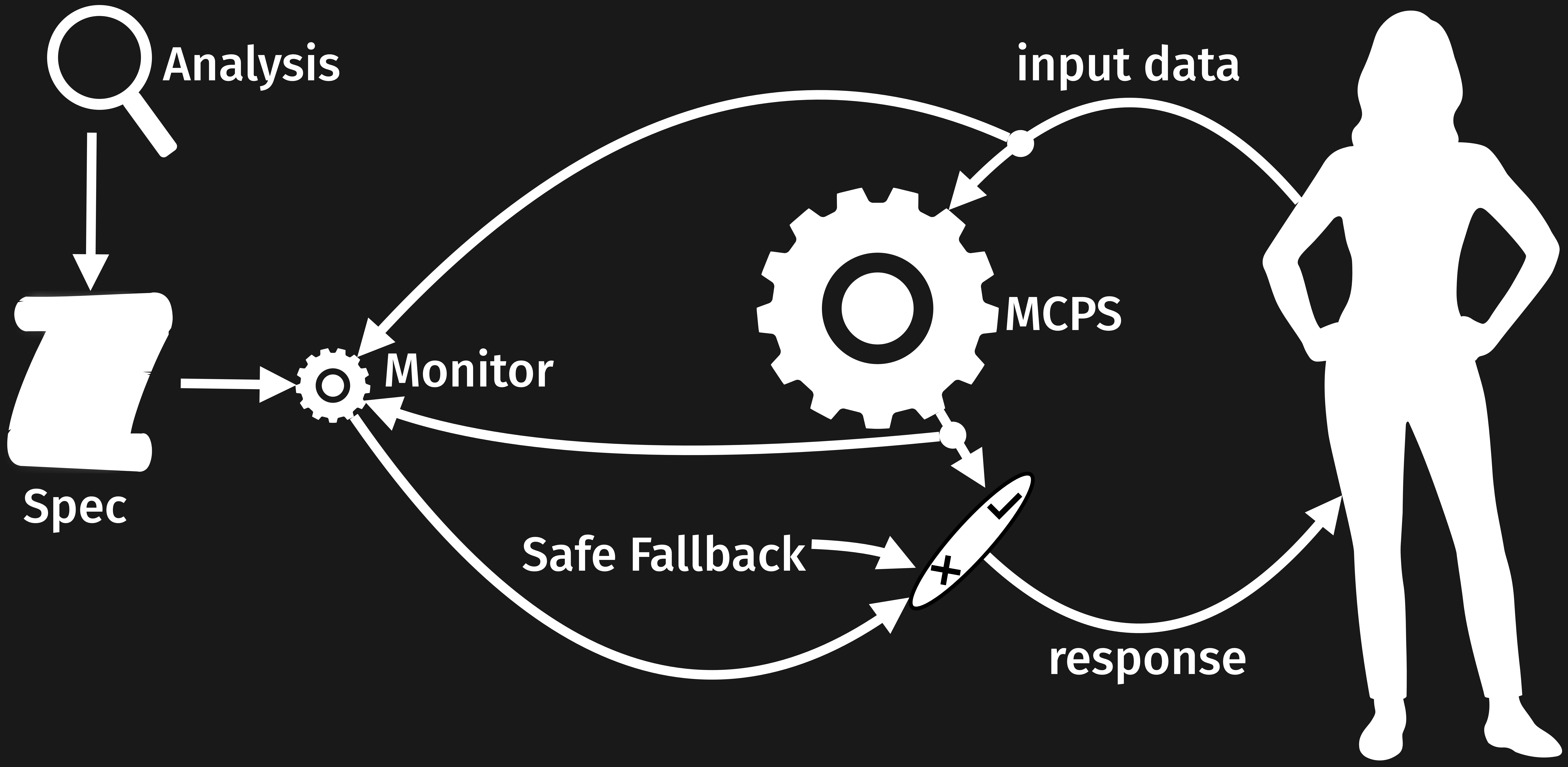
Monitoring MCPS



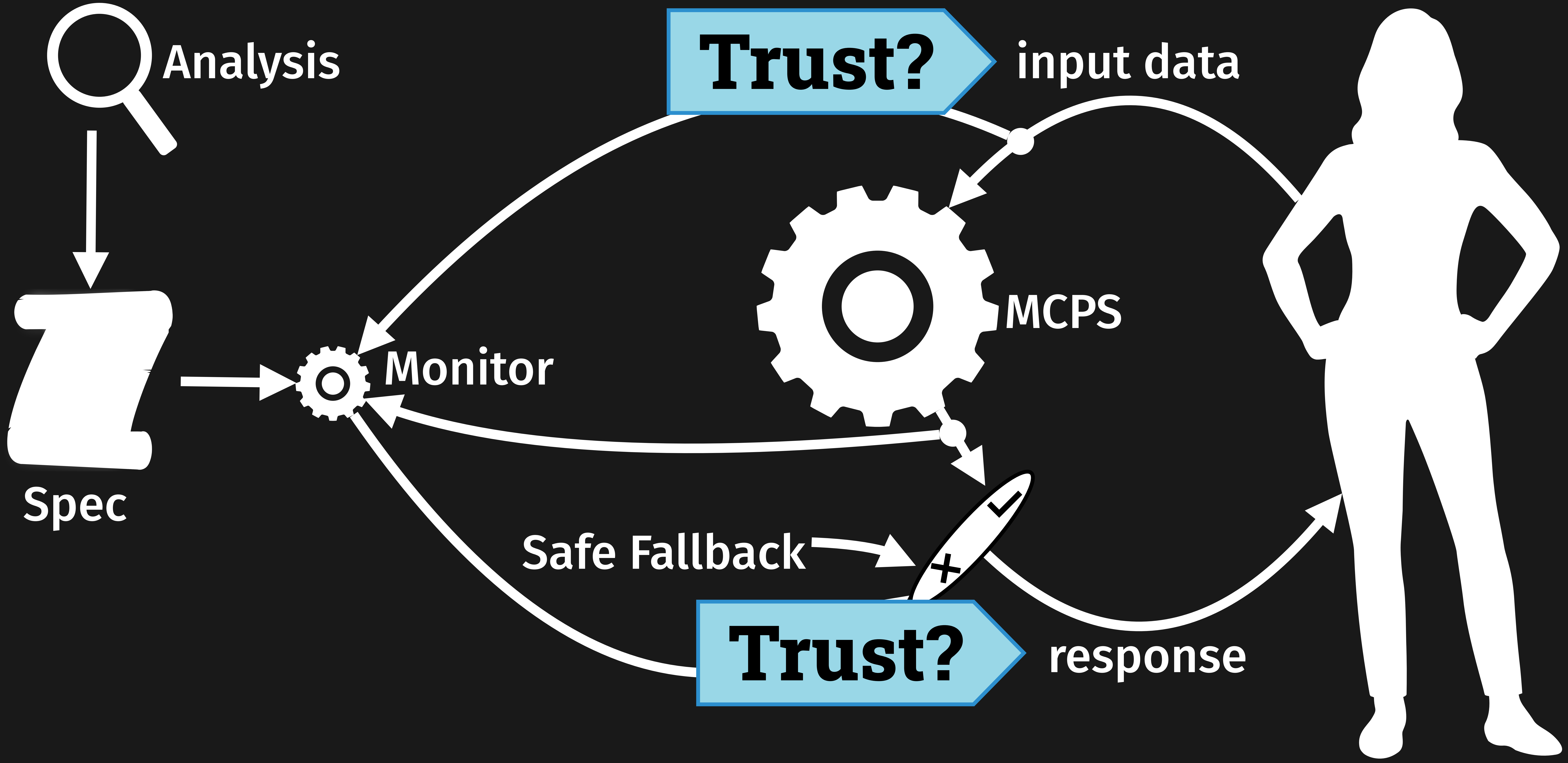
Monitoring MCPS



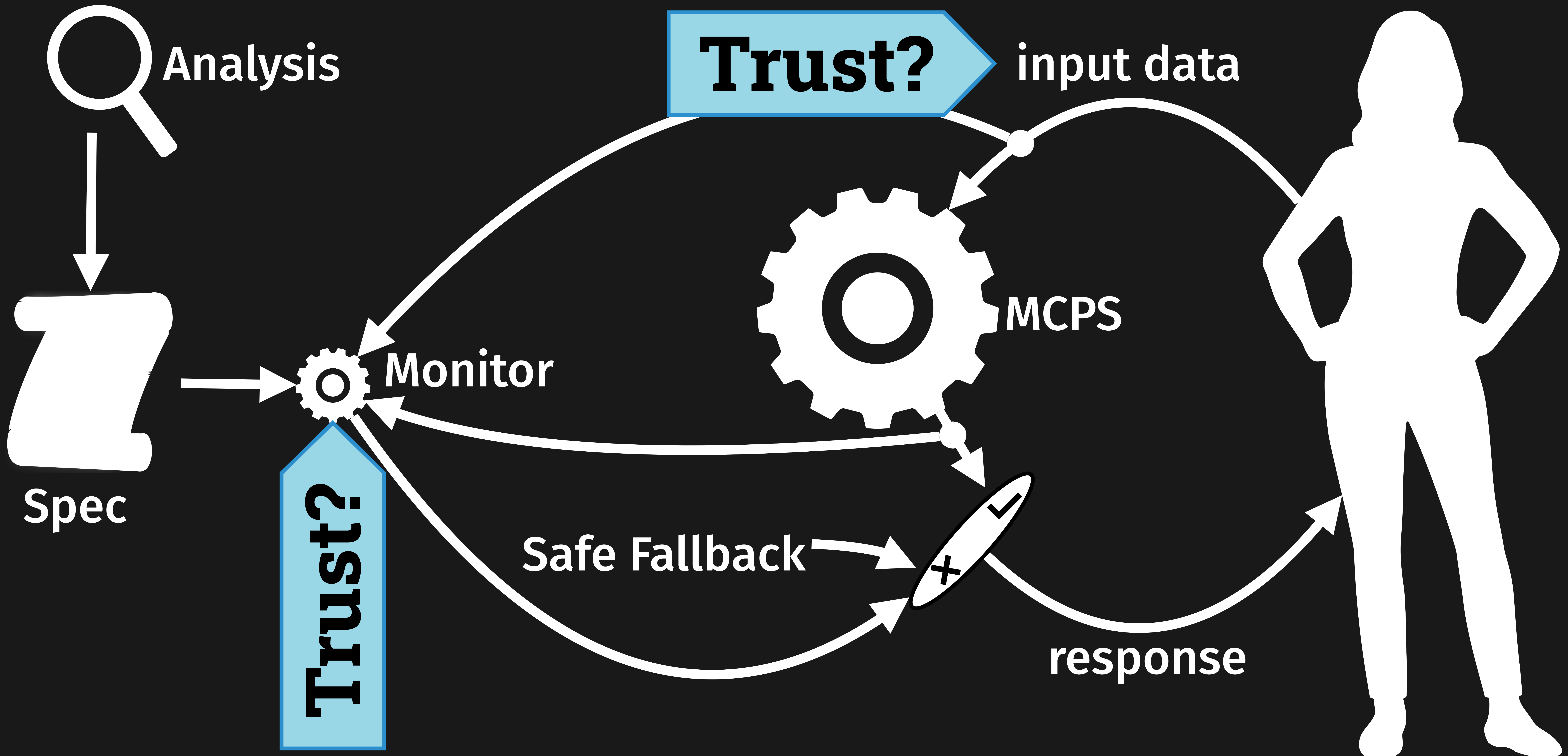
Monitoring MCPS



Monitoring MCPS



Monitoring MCPS



Runtime Monitoring Toolchains

ANALYZABLE

COMPREHENSIBLE

EXPRESSIVE

EFFICIENT

Runtime Monitoring Toolchains

ANALYZABLE

EXPRESSIVE

Lola

COMPREHENSIBLE

EFFICIENT

Real-Time-

DLR

Stream-based Monitoring

ANALYZABLE

COMPREHENSIBLE

EXPRESSIVE

EFFICIENT

Stream-based Monitoring

ANALYZABLE

EXPRESSIVE

COMPREHENSIBLE

EFFICIENT

Let's Have an Example

Let's Have an Example

```
input glucose: UInt
input admin_insulin: Bool
```

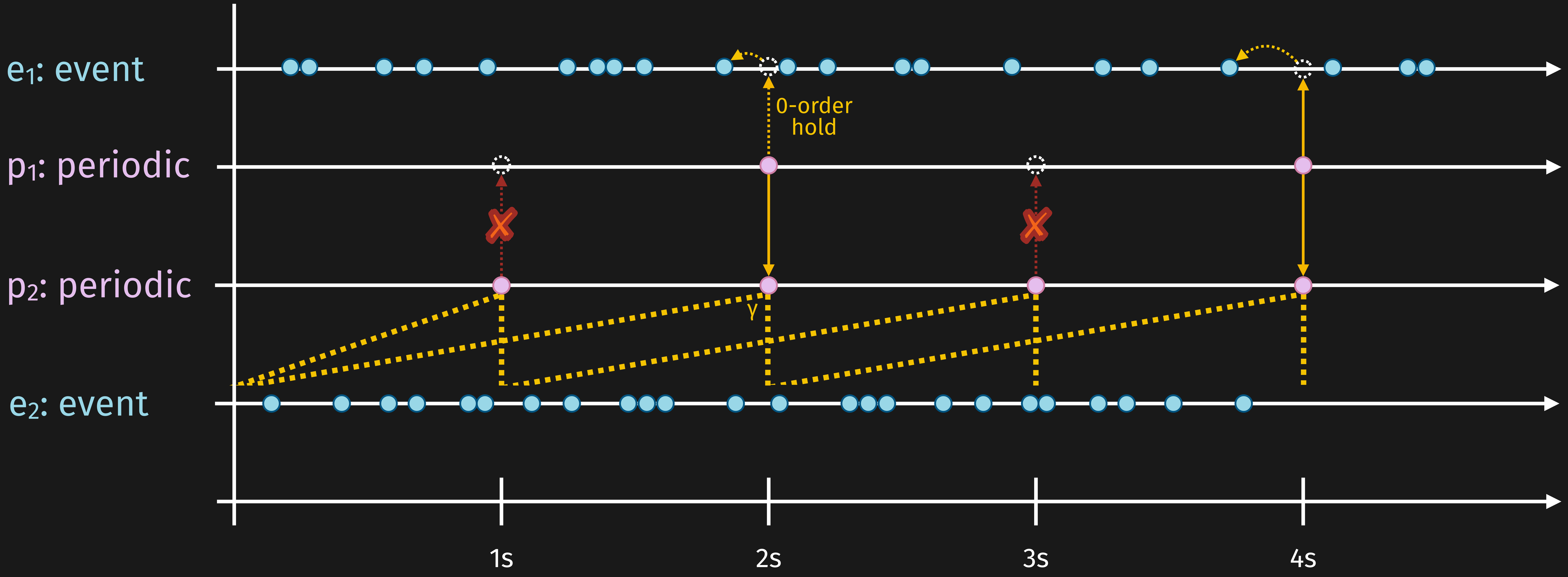
Let's Have an Example

```
input glucose: UInt
input admin_insulin: Bool

output clean_glucose := if glucose > 10 ∧ glucose < 300
                           then glucose else glucose.last(or: 90)
output admin_long  @1Hz   := admin_insulin.aggr(over: 10min, ∃)
output admin_short @100Hz := admin_insulin.aggr(over: 10sec, ∃)
```

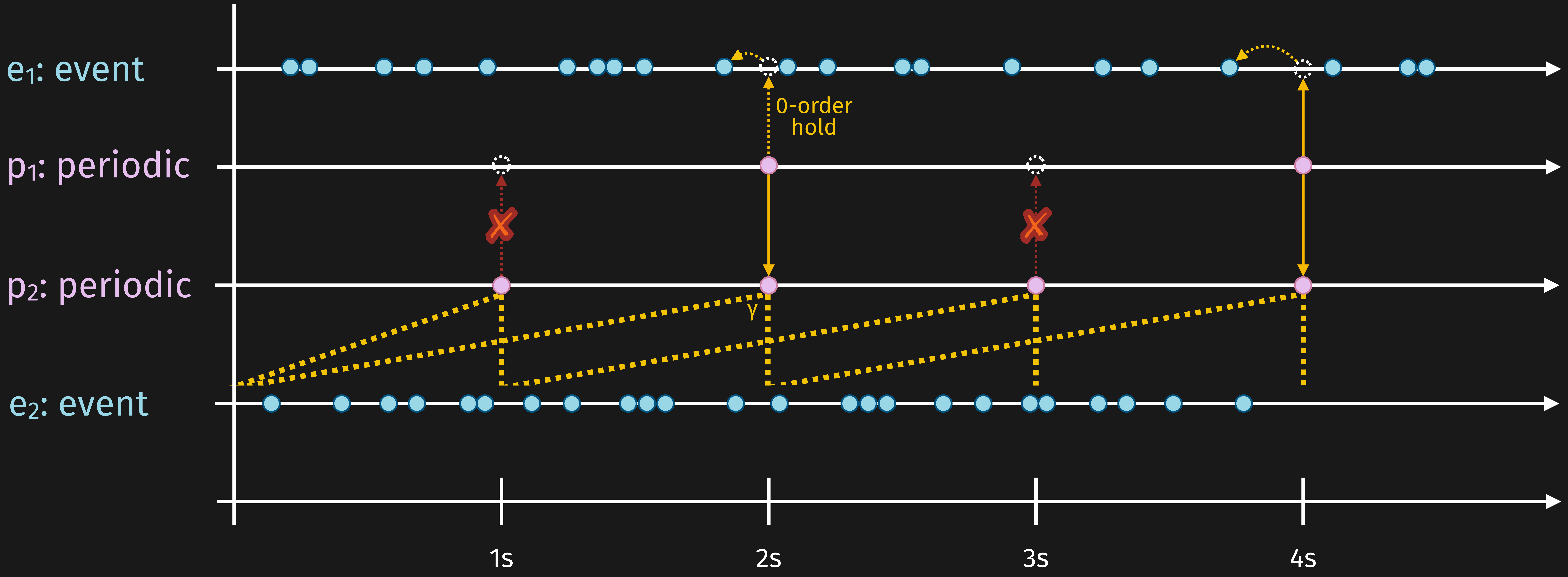
Let's Have an Example

```
input glucose: UInt
input admin_insulin: Bool

output clean_glucose := if glucose > 10 ∧ glucose < 300
                           then glucose else glucose.last(or: 90)
output admin_long  @1Hz    := admin_insulin.aggr(over: 10min, ∃)
output admin_short @100Hz := admin_insulin.aggr(over: 10sec, ∃)

trigger clean_glucose > 120 ∧ ¬admin_long.hold(or: ⊥)
                              “hyperglycemia untreated”
trigger clean_glucose < 60  ∧  admin_short.hold(or: ⊥)
                               “insulin despite hypoglycemia”
```

Quick Take: Two Notions of Time

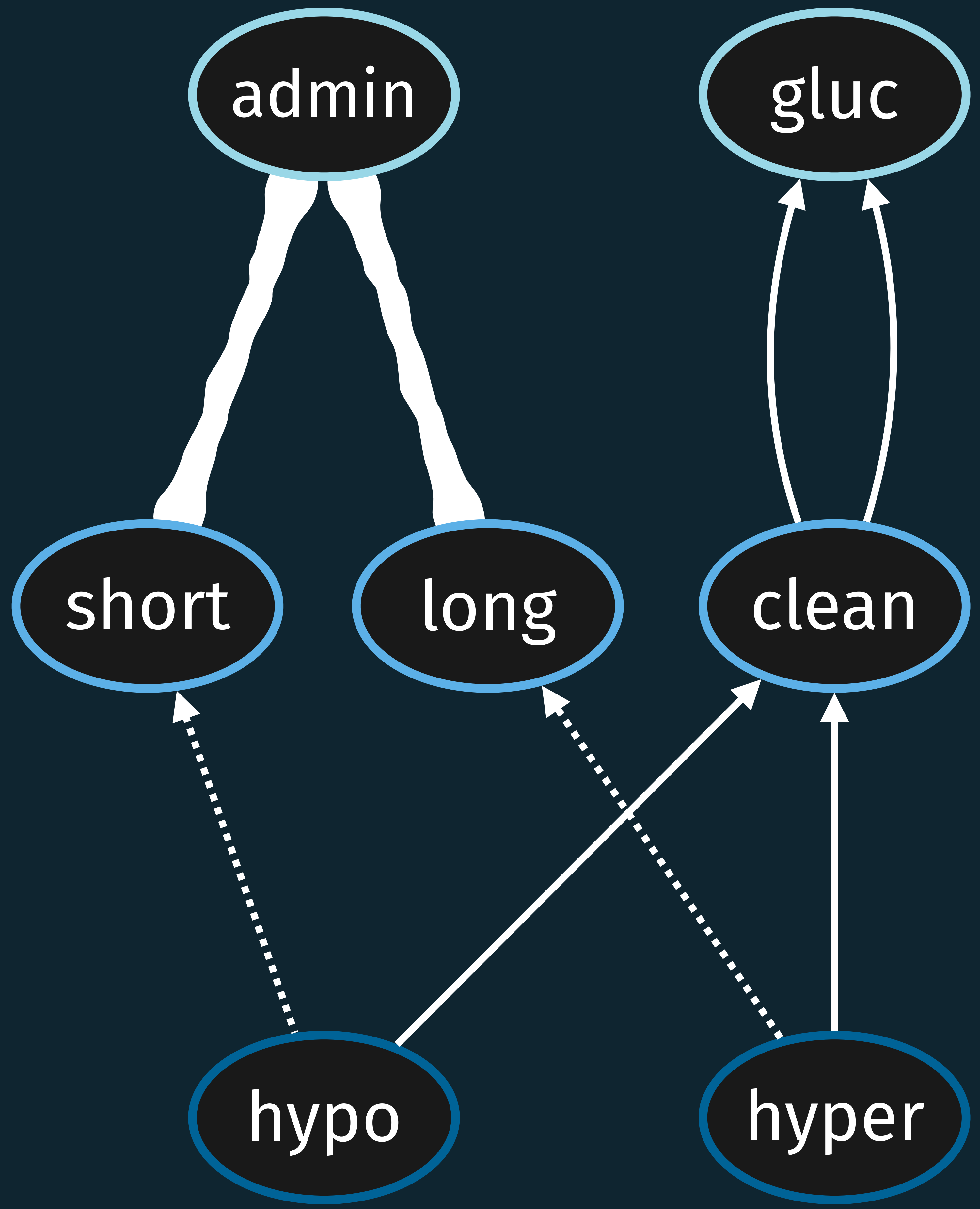


Quick Take: Two Notions of Time



👍 Checks for consistent timing in the spec.

Specification Analysis



hyperglycemia

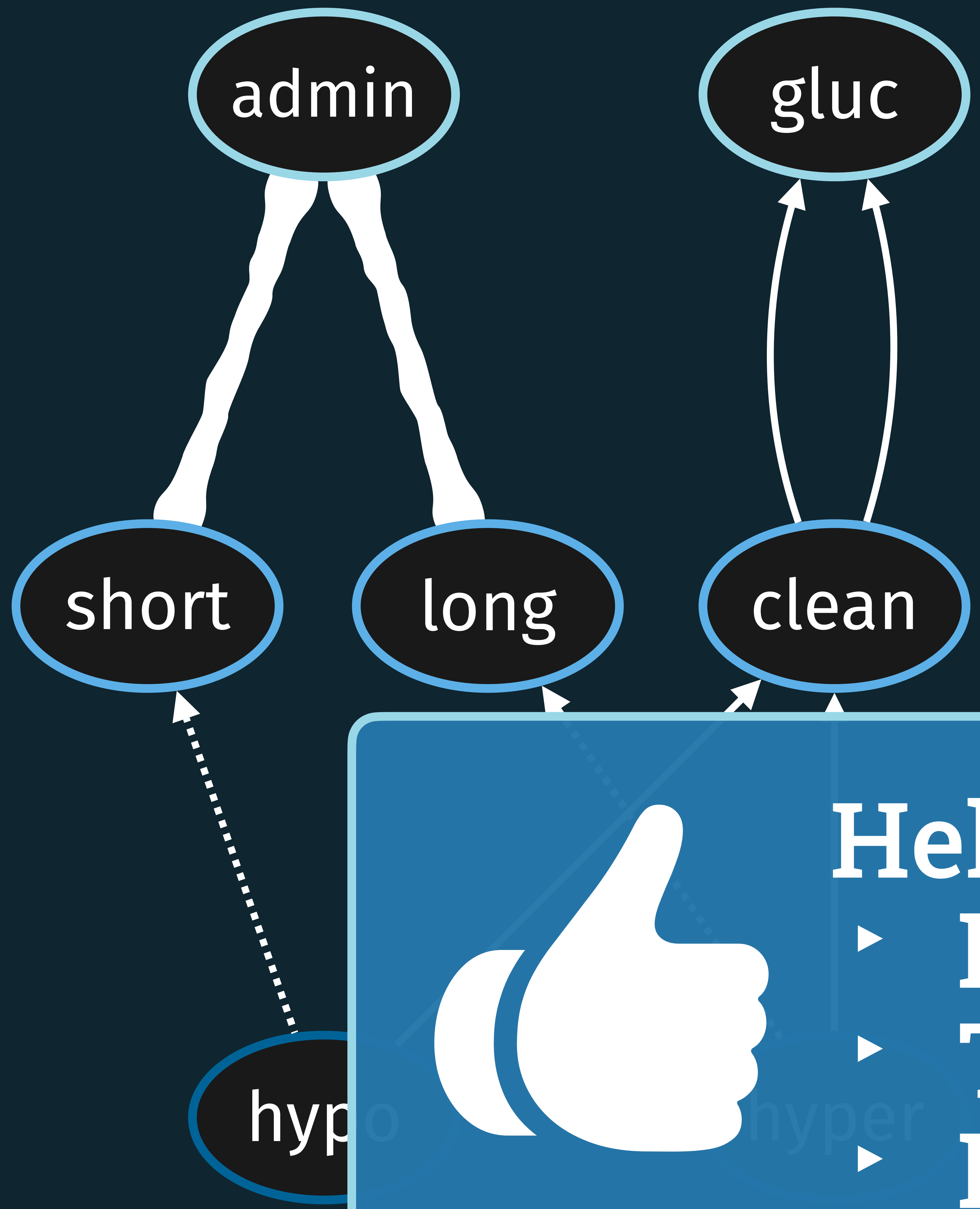
Memory Footprint: 5B

Timing: @{glucose}

Concurrent Eval: #2

Sequential Eval: #4

Specification Analysis



⚠ **hyperglycemia**

Memory Footprint: 5B

Timing: @{glucose}

Concurrent Eval: #2

Sequential Eval: #4

Helps specifiers to understand:

- **Resource Consumption**
- **Timing Behavior**
- **Running Time**

Quick Take: Efficiency

Quick Take: Efficiency

Interpreter	438ns	1.535µs
Compilation	6ns	63ns

Quick Take: Efficiency

Interpreter	438ns	1.535µs
Compilation	6ns	63ns

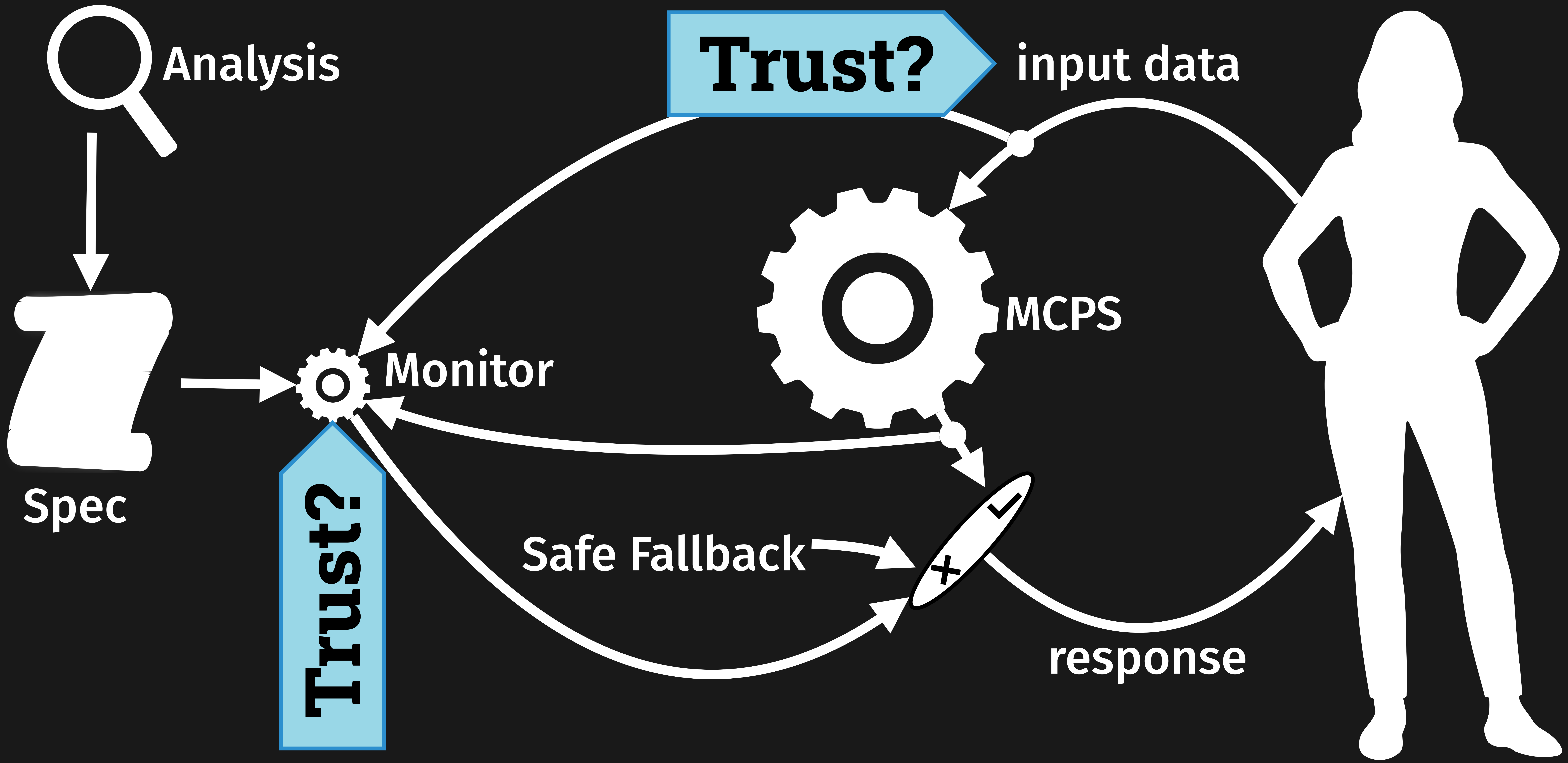
ANALYZABLE

EXPRESSIVE

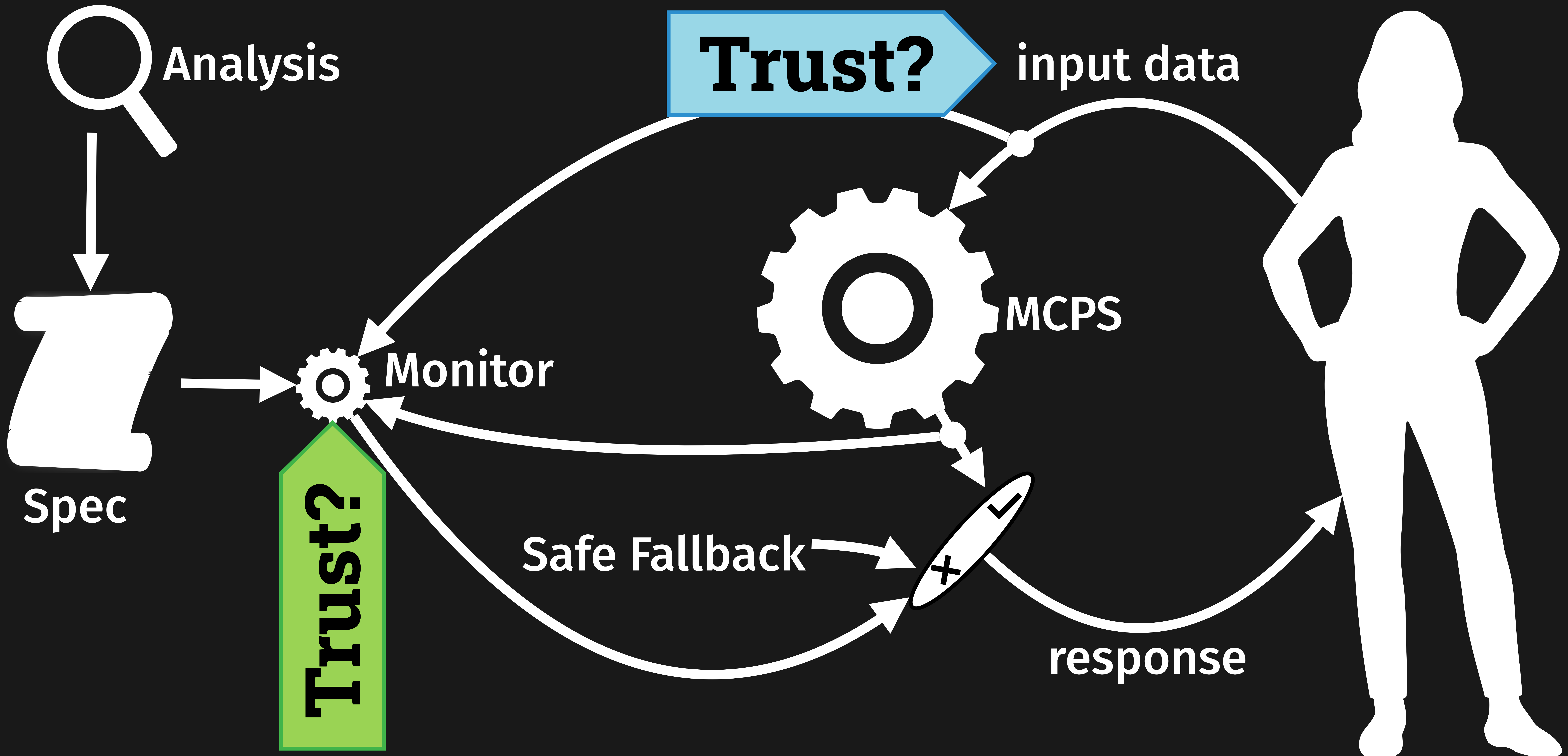
COMPREHENSIBLE

EFFICIENT

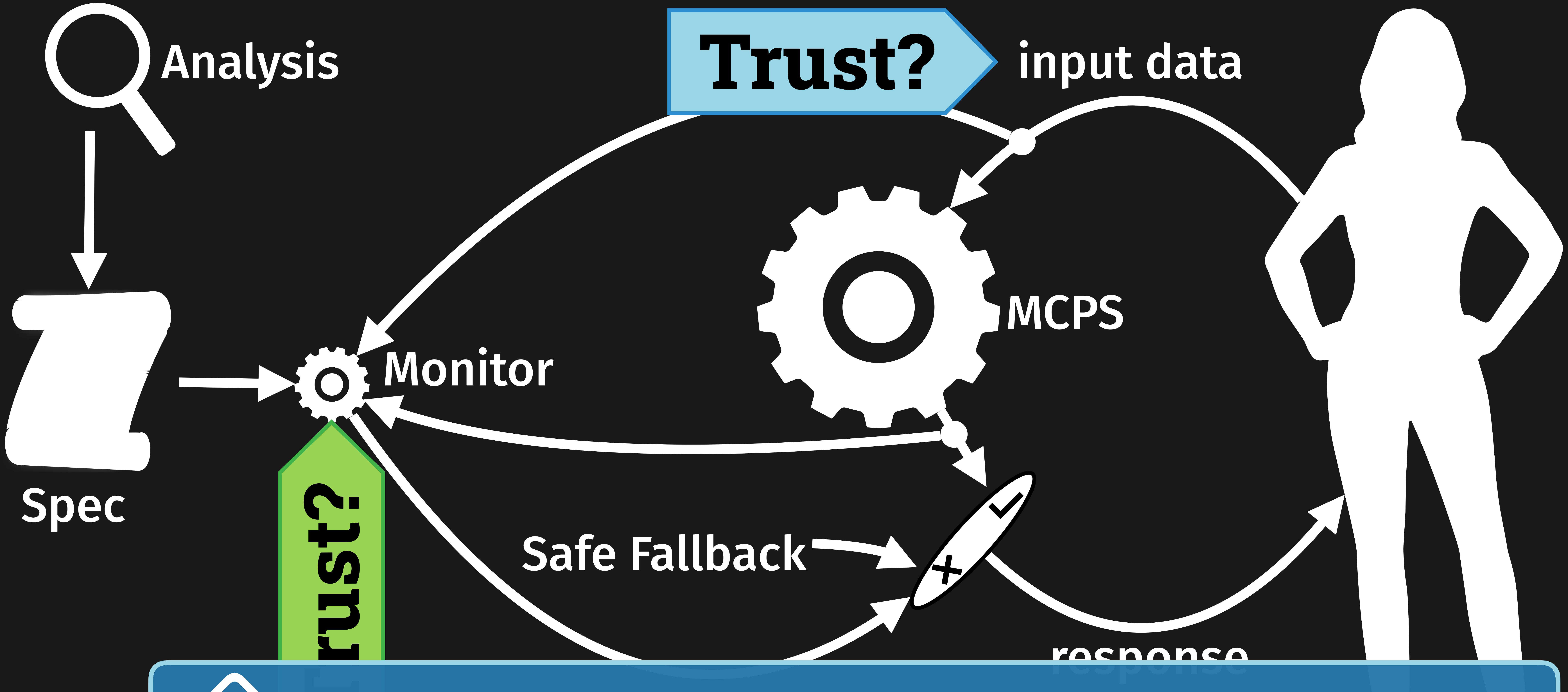
Monitoring MCPS



Monitoring MCPS



Monitoring MCPS



? How dependent on the input is the monitor?

Robustness

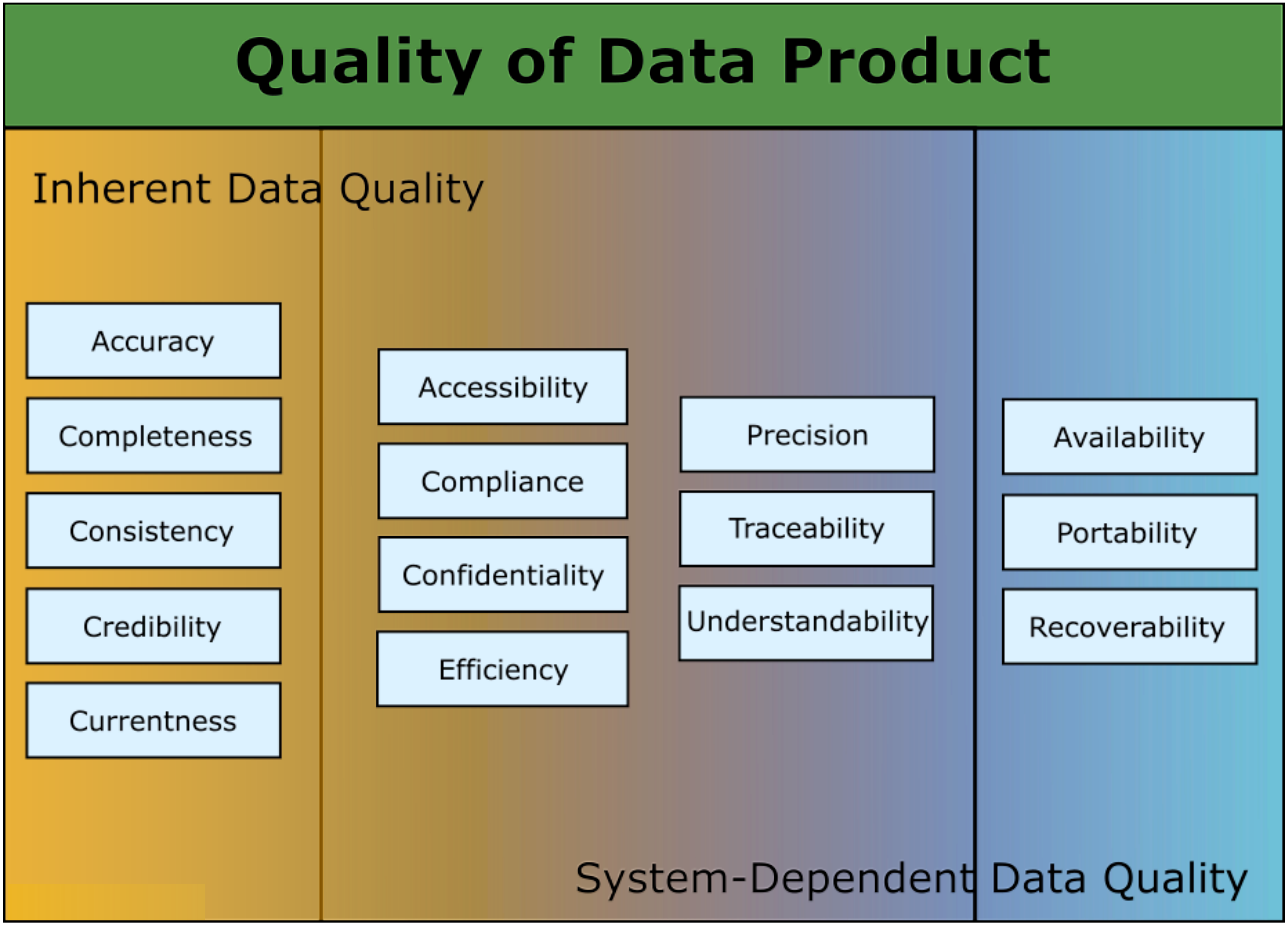
.810
RTLola
RTLola
.812

Robustness

A system is ***robust*** iff

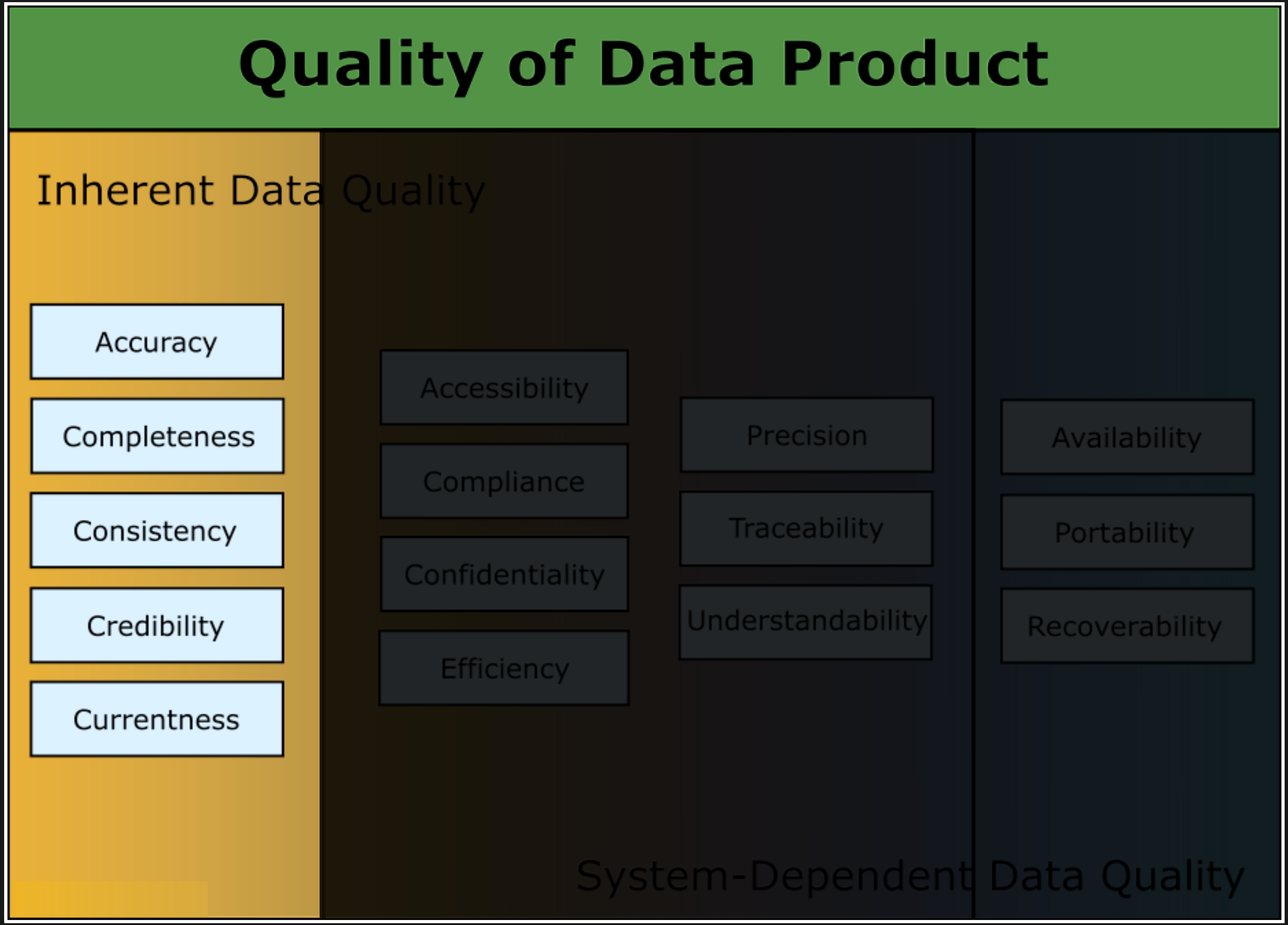
minor input *deviation* → *minor* output *deviation*

Background: Deviation Criteria



Provided by iso25000.com → ISO 25012 (thanks to Reviewer #1)

Background: Deviation Criteria



Quality of Data Product

Inherent Data Quality

Accuracy

Completeness

Consistency

Credibility

Currentness

Accessibility

Compliance

Confidentiality

Efficiency

Precision

Traceability

Understandability

Availability

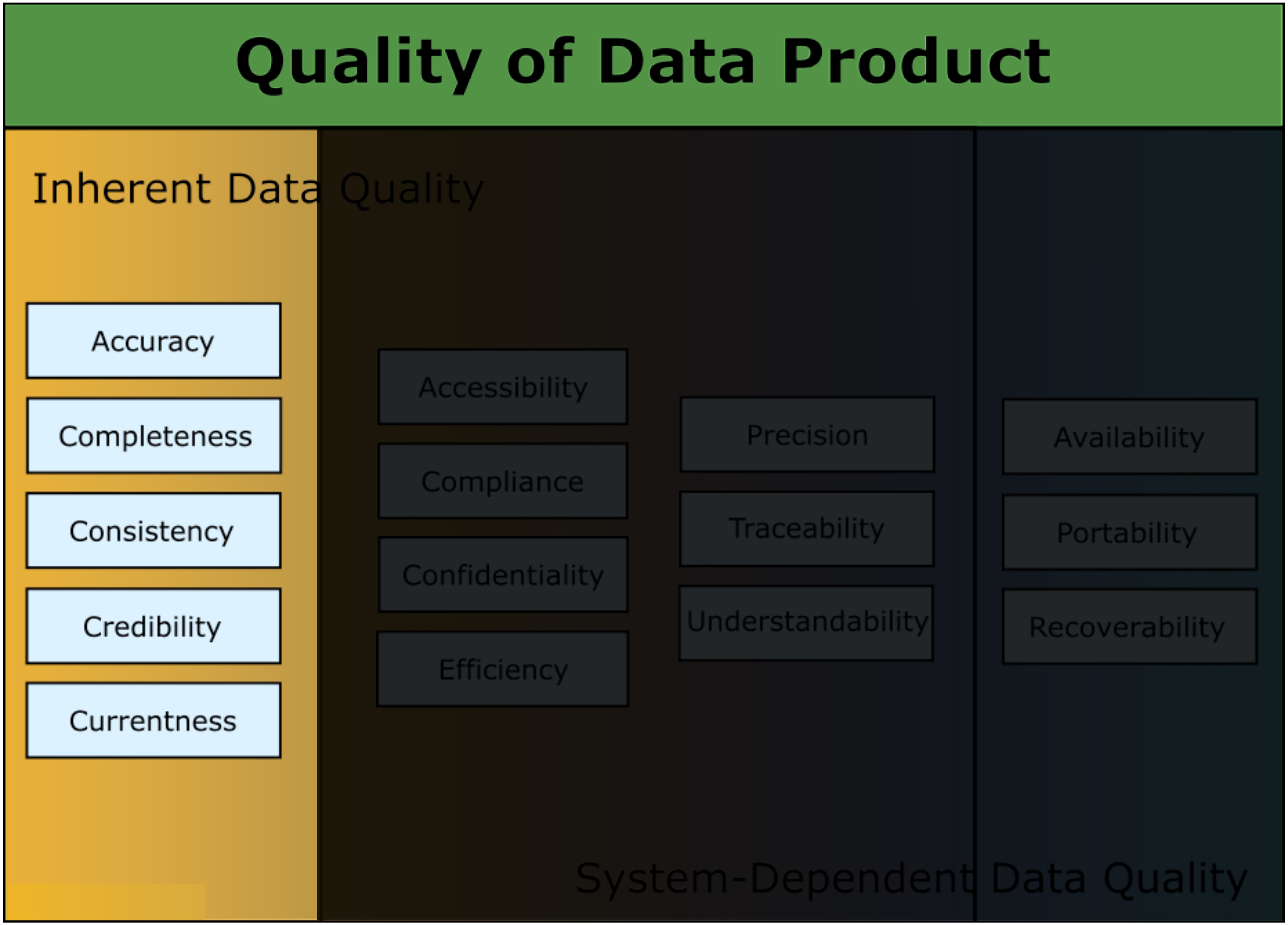
Portability

Recoverability

System-Dependent Data Quality

Provided by iso25000.com → ISO 25012 (thanks to Reviewer #1)

BACKGROUND: DEVIATION CRITERIA



Quality of Data Product

Inherent Data Quality

Accuracy

Completeness

Consistency

Credibility

Currentness

Accessibility

Compliance

Confidentiality

Efficiency

Precision

Traceability

Understandability

Availability

Portability

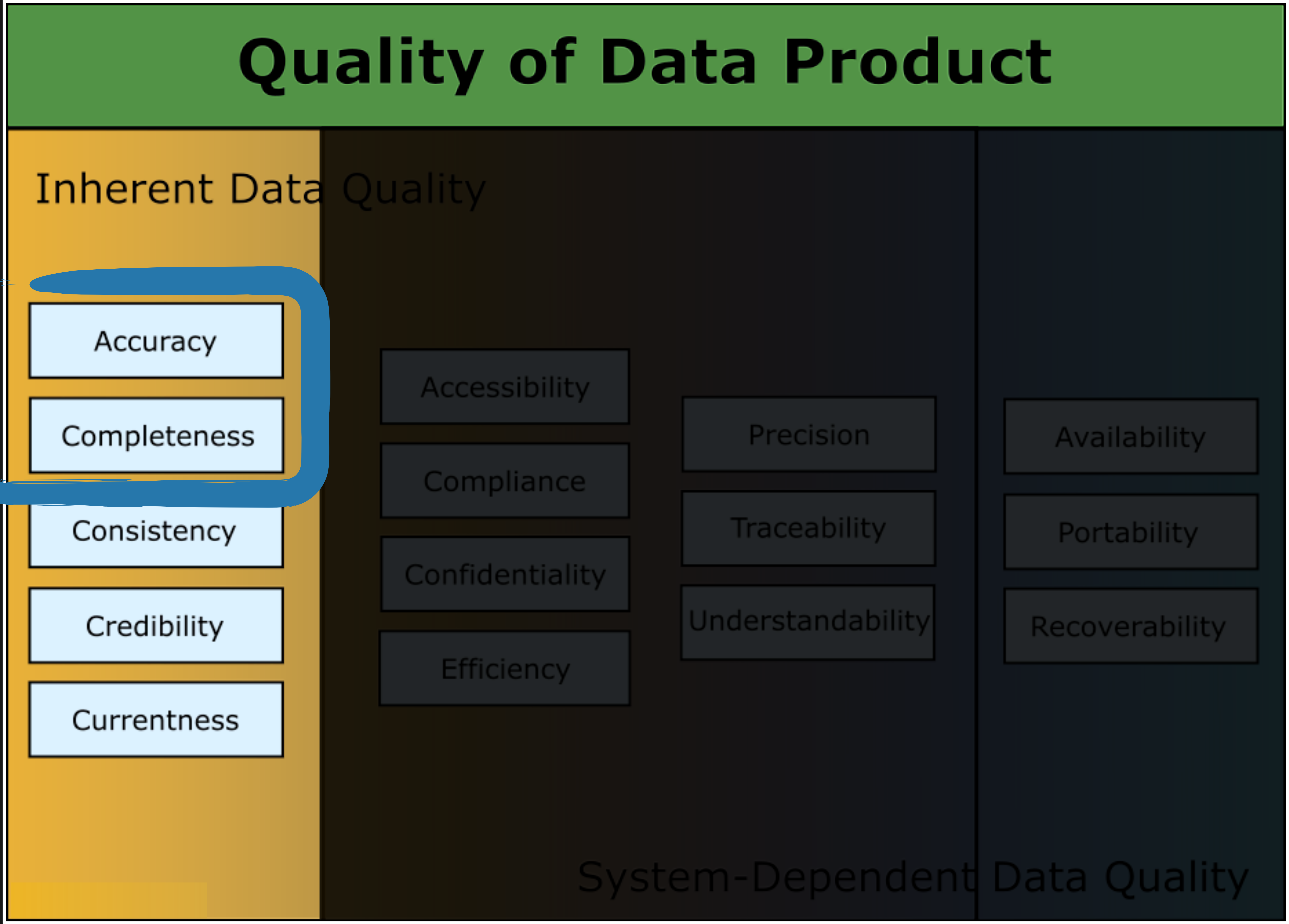
Recoverability

System-Dependent Data Quality

part of spec

Provided by iso25000.com → ISO 25012 (thanks to Reviewer #1)

Background: Deviation Criteria



Quality of Data Product

Inherent Data Quality

Accuracy

Completeness

Consistency

Credibility

Currentness

Accessibility

Compliance

Confidentiality

Efficiency

Precision

Traceability

Understandability

Availability

Portability

Recoverability

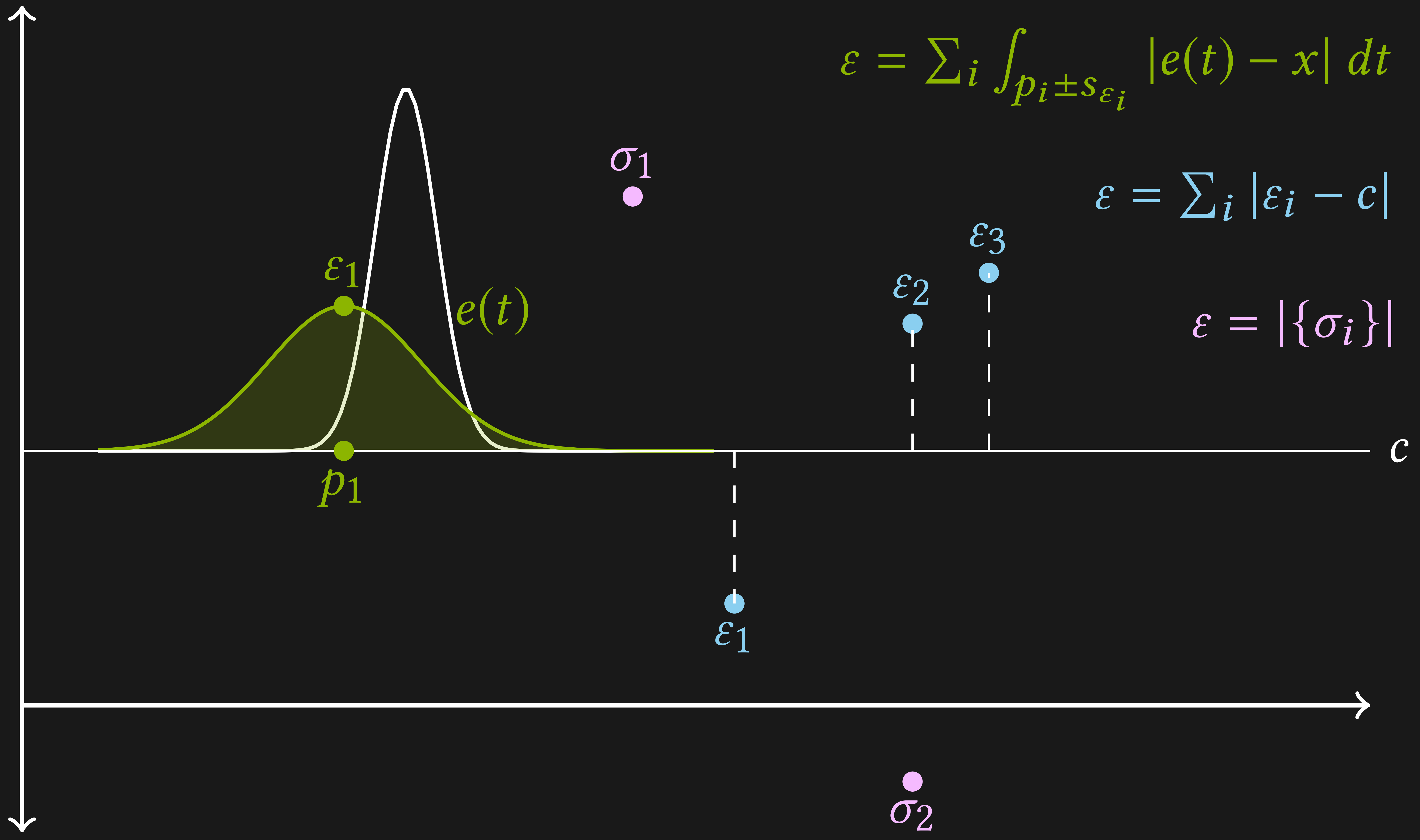
System-Dependent Data Quality

part of spec

Provided by iso25000.com → ISO 25012 (thanks to Reviewer #1)

Error Models: Accuracy

A system is ***robust*** iff
minor input *deviation* → *minor* output *deviation*

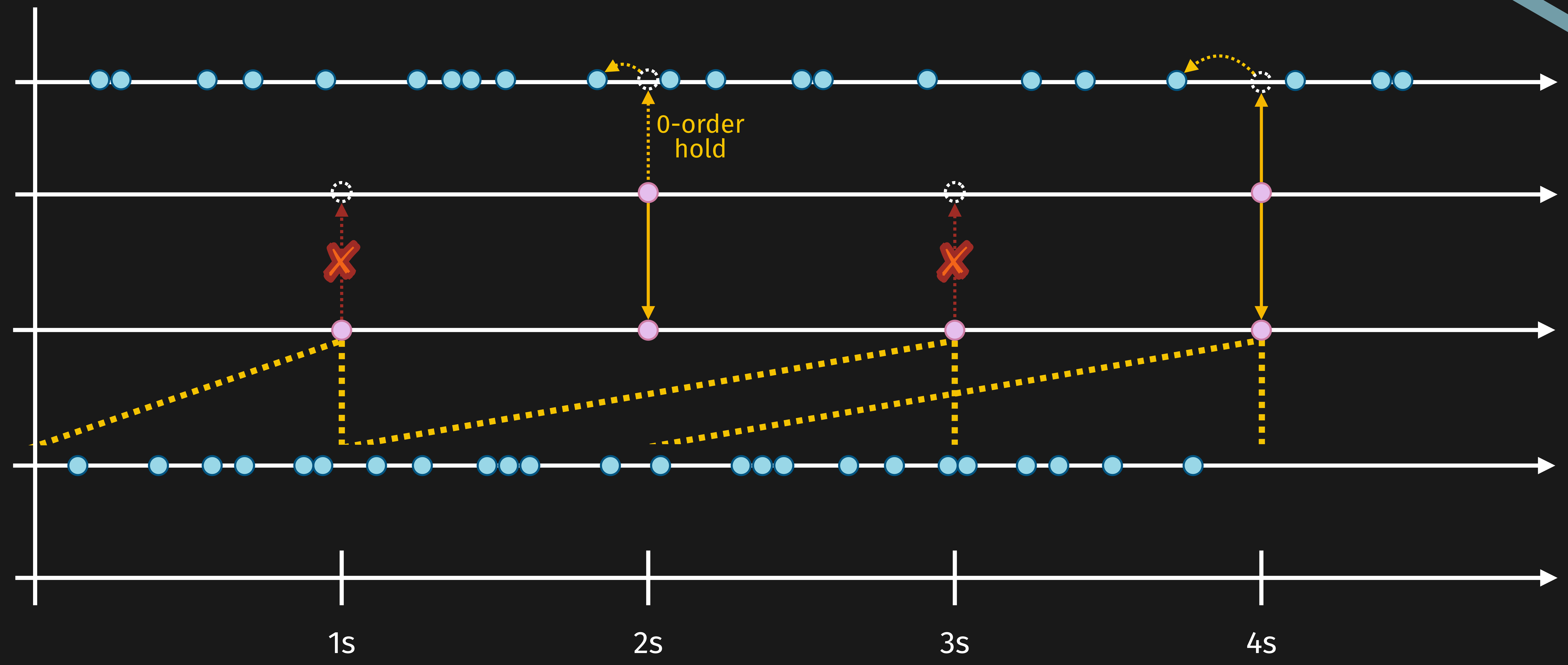


$$\varepsilon = \sum_i \int_{p_i \pm s_{\varepsilon_i}} |e(t) - x| \, dt$$

$$\varepsilon = \sum_i |\varepsilon_i - c|$$

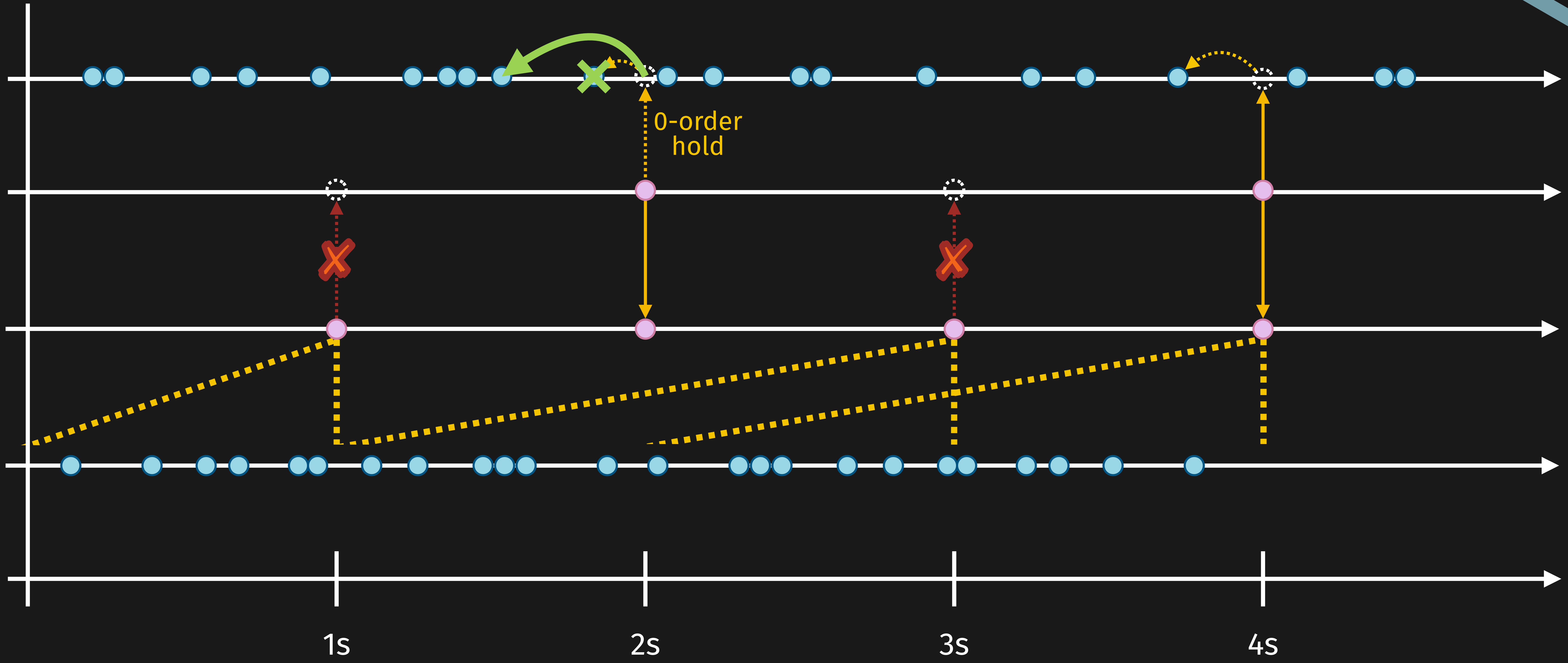
$$\varepsilon = |\{\sigma_i\}|$$

Error Models: Completeness

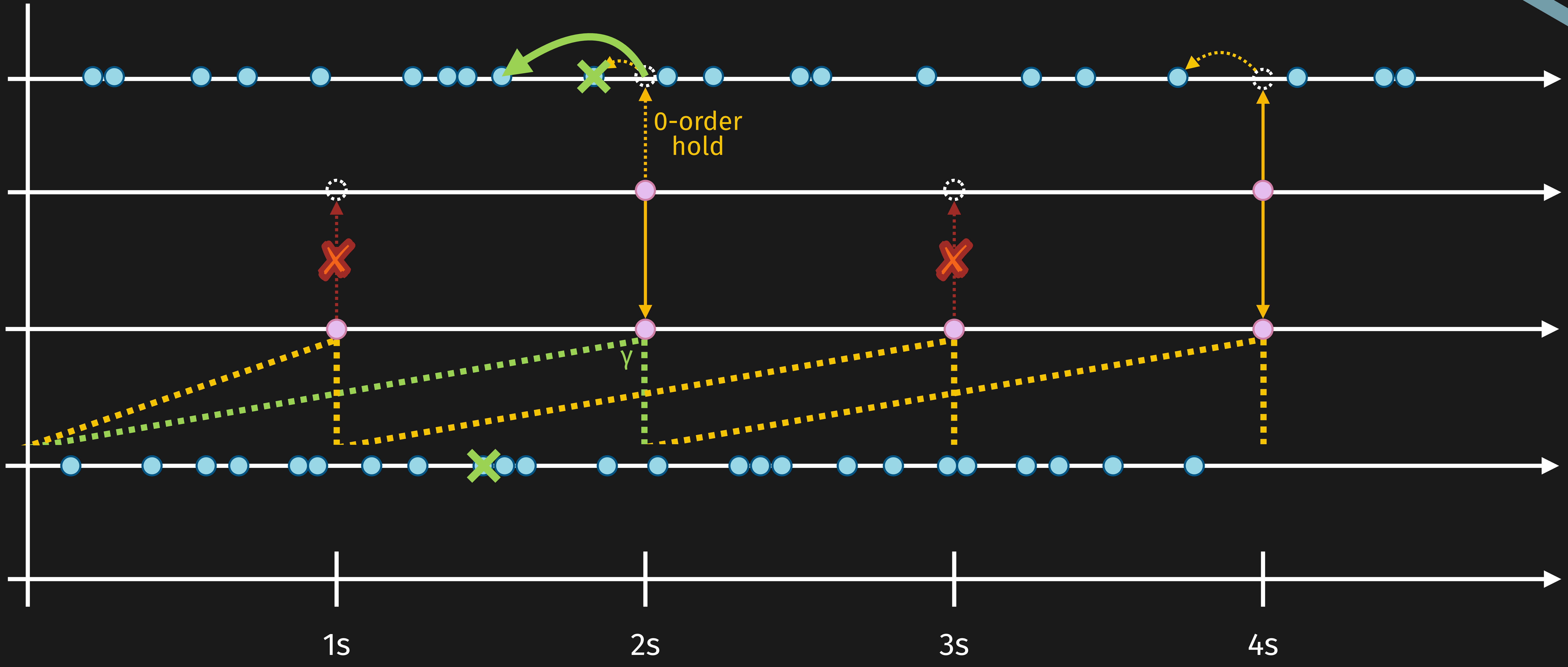


Error Models: Completeness

WIP

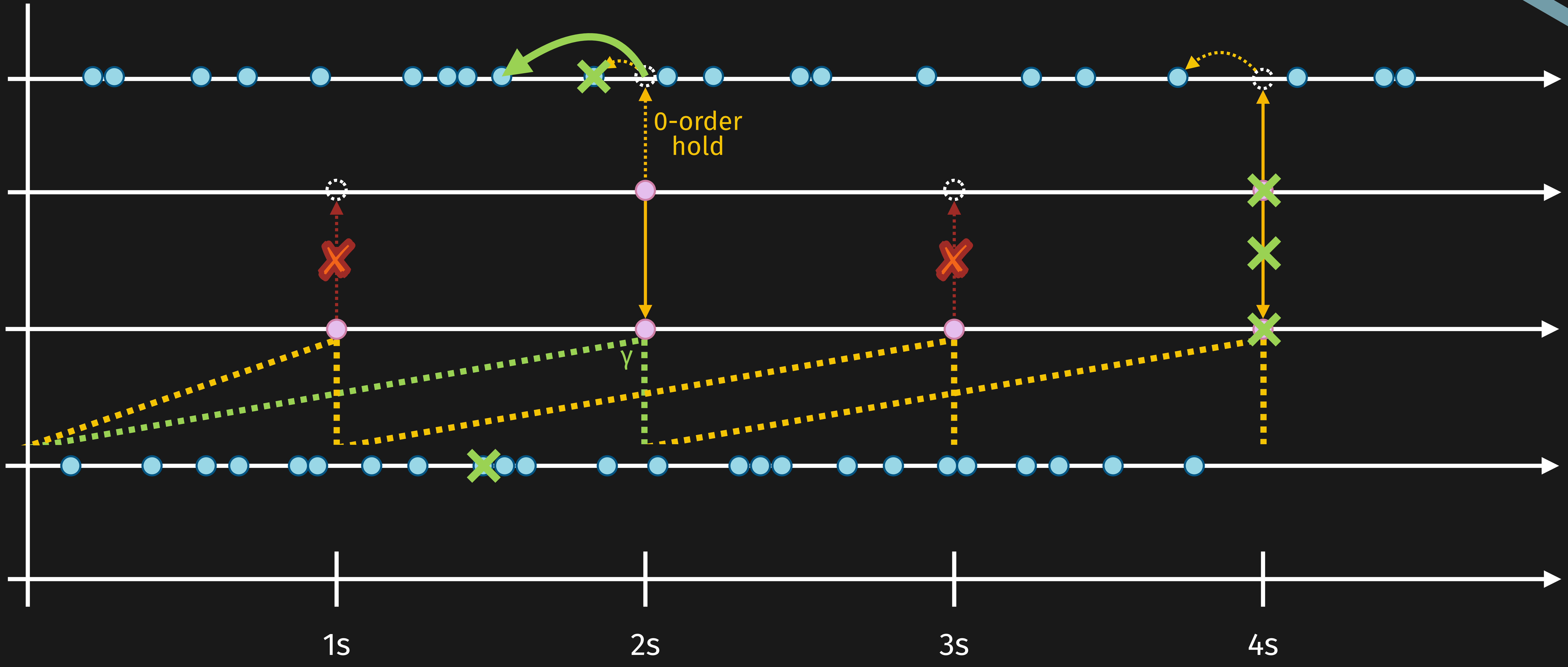


Error Models: Completeness



Error Models: Completeness

WIP



Robustness: Definition

A *monitor* is ε-δ-***robust*** iff

minor input deviation → minor output deviation

$$\forall v, \overline{v} \colon dist(v, \overline{v}) \leq \varepsilon \implies dist(M(v), M(\overline{v})) \leq \delta$$

Robustness: Definition

$$\forall v, \overline{v} \colon dist(v, \overline{v}) \leq \varepsilon \implies dist(M(v), M(\overline{v})) \leq \delta$$

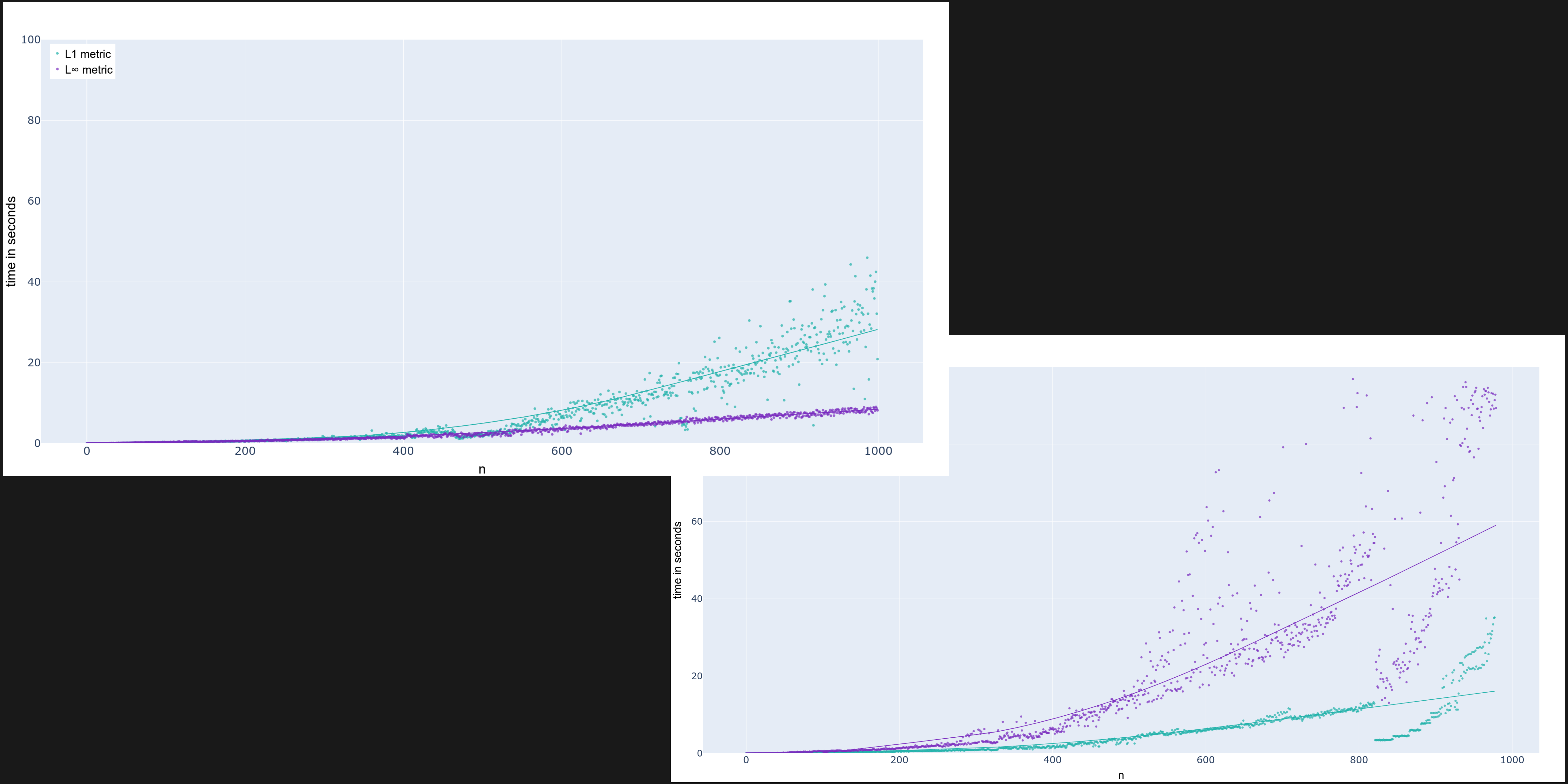
$\max \delta$ **s.t.**

$$\sum_{i \in in(\Phi)} d(v^{s_i}, \overline{v}^{s_i}) \leq \varepsilon \qquad \sum_{i \in out(\Phi)} d(v^{s_i}, \overline{v}^{s_i}) = \delta$$

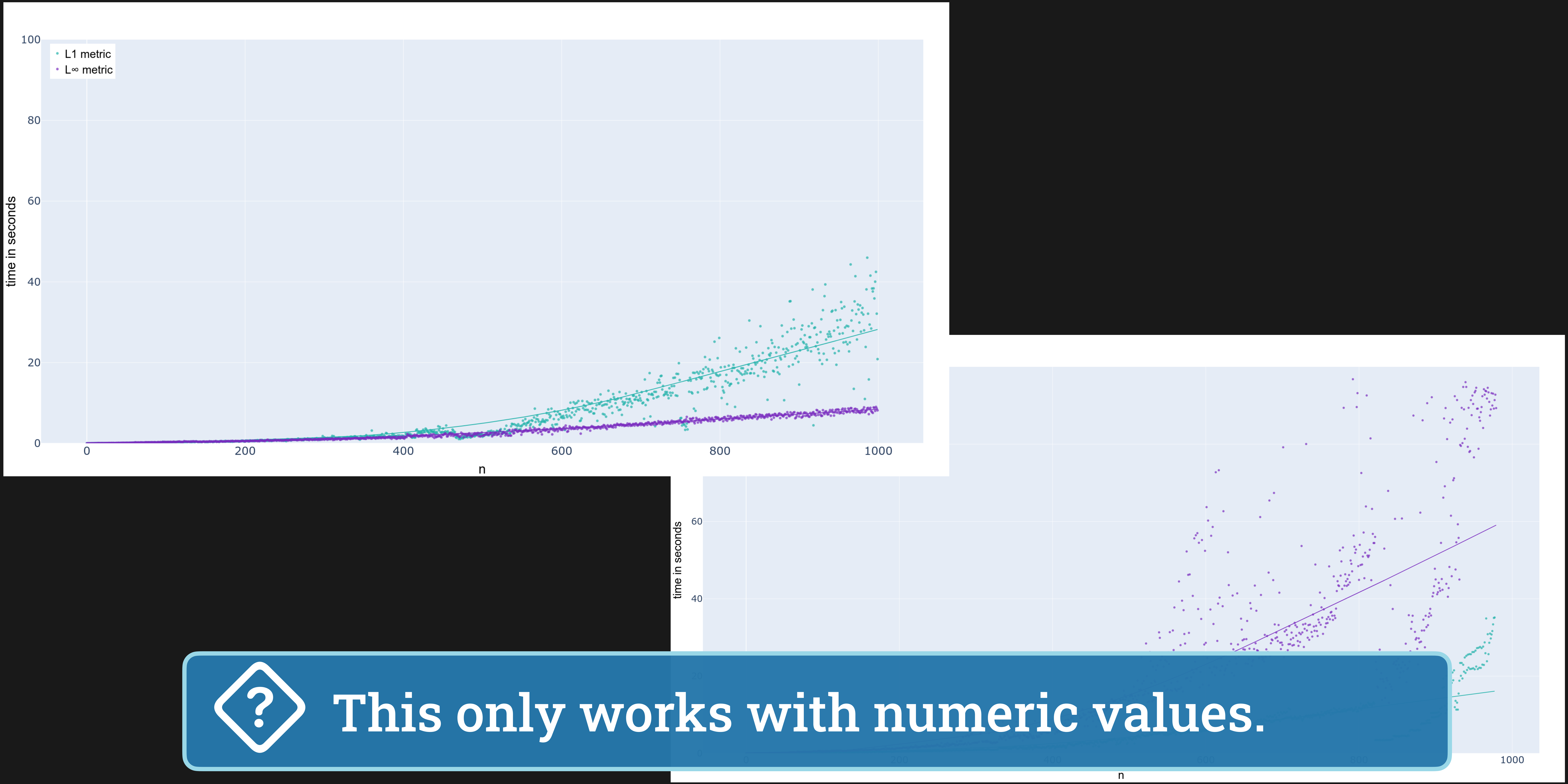
$$\bigwedge_{i \in out(\Phi)} \bigwedge_{1 \leq \eta \leq n} \overline{v}^{s_i}_{\eta} = \overline{enc}_{\eta}(s_i) \qquad \bigwedge_{i \in out(\Phi)} \bigwedge_{1 \leq \eta \leq n} v^{s_i}_{\eta} = enc_{\eta}(s_i)$$

Fix point computation over length of traces

Evaluation


100
80
60
40
20
0
time in seconds
L1 metric
L∞ metric
0
200
400
600
800
1000
n
60
40
20
0
time in seconds
0
200
400
600
800
1000
n

Evaluation



This only works with numeric values.

Non-Numeric Values

WIP

```
input glucose: UInt
input admin_insulin: Bool

output clean_glucose := if glucose > 10 ∧ glucose < 300
                           then glucose else glucose.last(or: 90)
output admin_long  @1Hz   := admin_insulin.aggr(over: 10min, ∃)
output admin_short @100Hz := admin_insulin.aggr(over: 10sec, ∃)

trigger glucose > 120 ∧ ¬admin_long.hold(or: ⊥)
                       “hyperglycemia untreated”
trigger glucose < 60  ∧ admin_short.hold(or: ⊥)
                       “insulin despite hypoglycemia”
```

Non-Numeric Values

WIP

```
input glucose: UInt
input admin_insulin: Bool

output clean_glucose := if glucose > 10 ∧ glucose < 300
                           then glucose else glucose.last(or: 90)
output admin_long  @1Hz   := admin_insulin.aggr(over: 10min, ∃)
output admin_short @100Hz := admin_insulin.aggr(over: 10sec, ∃)

trigger glucose > 120 ∧ ¬admin_long.hold(or: ⊥)
                       “hyperglycemia untreated”
trigger glucose < 60  ∧ admin_short.hold(or: ⊥)
                       “insulin despite hypoglycemia”
```

Non-Numeric Values

WIP

```
input glucose: UInt
input admin_insulin: Bool

output clean_glucose := if glucose > 10 ∧ glucose < 300
                           then glucose else glucose.last(or: 90)
output admin_long  @1Hz   := admin_insulin.aggr(over: 10min, ∃)
output admin_short @100Hz := admin_insulin.aggr(over: 10sec, ∃)

trigger glucose > 120 ∧ ¬admin_long.hold(or: ⊥)
                       “hyperglycemia untreated”
trigger glucose < 60  ∧ admin_short.hold(or: ⊥)
                       “insulin despite hypoglycemia”
```

⚡ **Some language constructs are brittle by design**

Non-Numeric Values

WIP

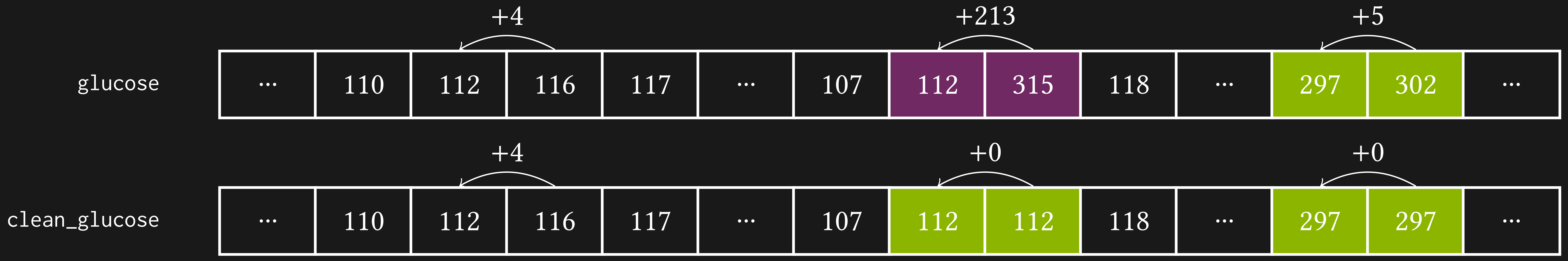
```
input glucose: UInt
input admin_insulin: Bool

output clean_glucose := if glucose > 10 ∧ glucose < 300
                           then glucose else glucose.last(or: 90)
output admin_long  @1Hz   := admin_insulin.aggr(over: 10min, ∃)
output admin_short @100Hz := admin_insulin.aggr(over: 10sec, ∃)

trigger glucose > 120 ∧ ¬admin_long.hold(or: ⊥)
                        “hyperglycemia untreated”
trigger glucose < 60  ∧ admin_short.hold(or: ⊥)
                        “insulin despite hypoglycemia”
```

⚡ Some language constructs are brittle by design

Non-Numeric Values

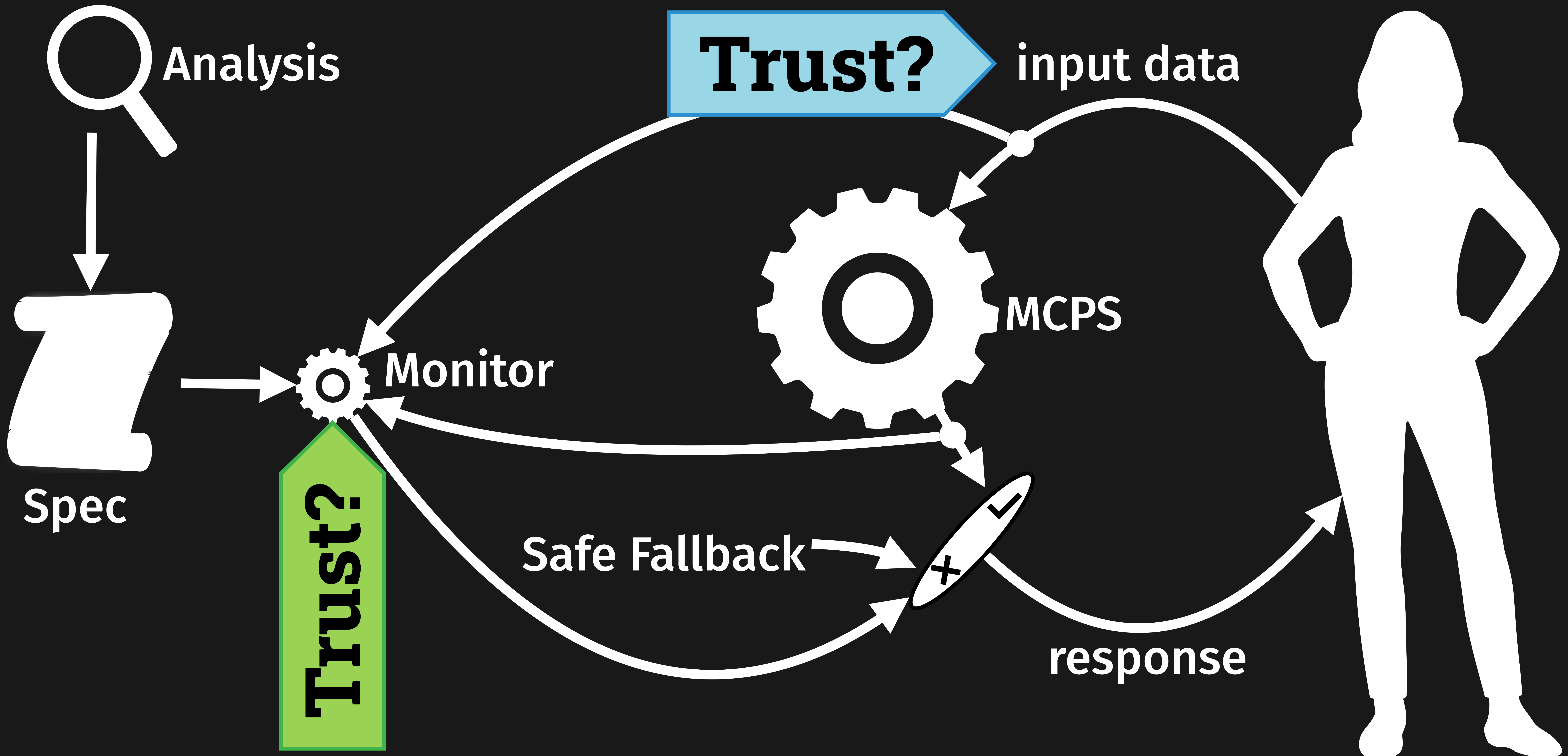
WIP

```
input glucose: UInt

output clean_glucose := if glucose < 300
                           then glucose else glucose.last(or: 90)

trigger glucose > 150 “hyperglycemia untreated”
```

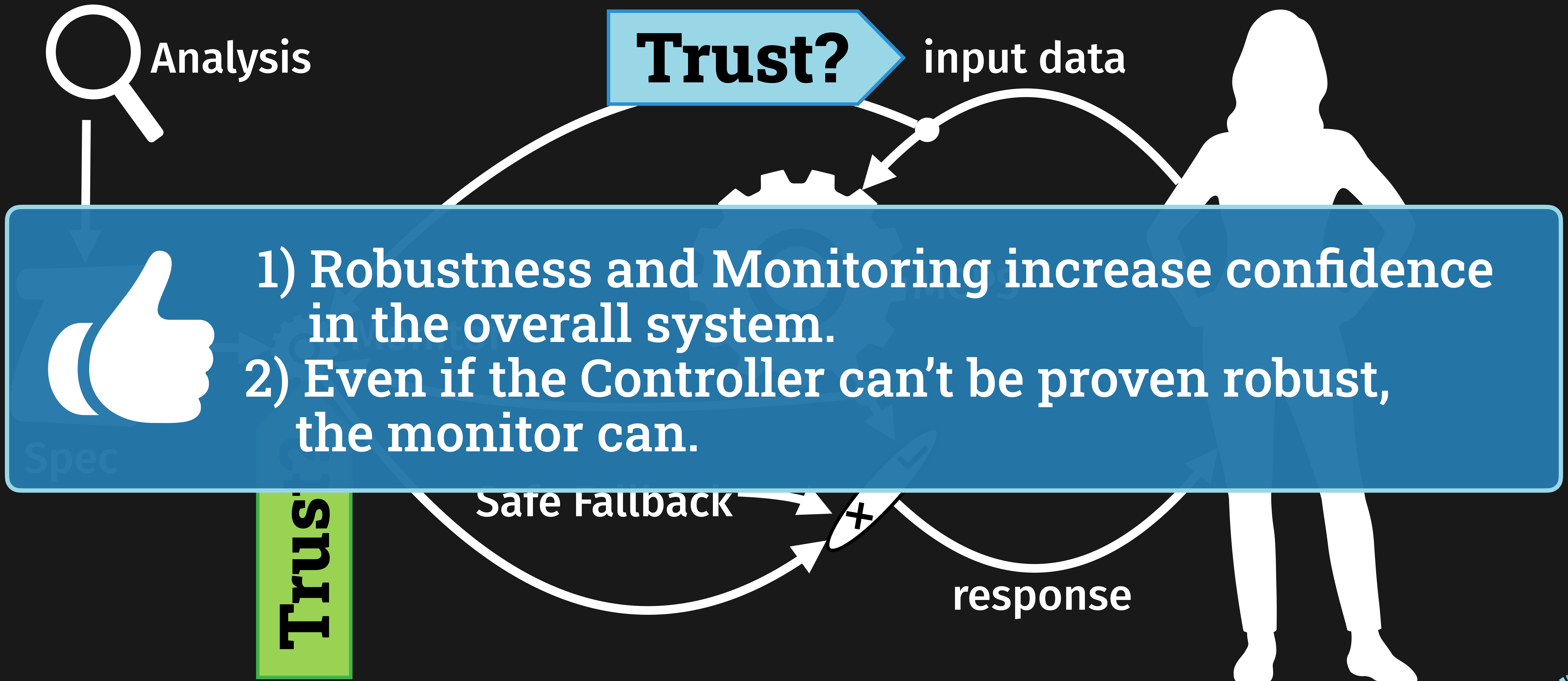


glucose	…	110	112	116	117	…	107	112	315	118	…	297	302	…
clean_glucose	…	110	112	116	117	…	107	112	112	118	…	297	297	…

Conclusion

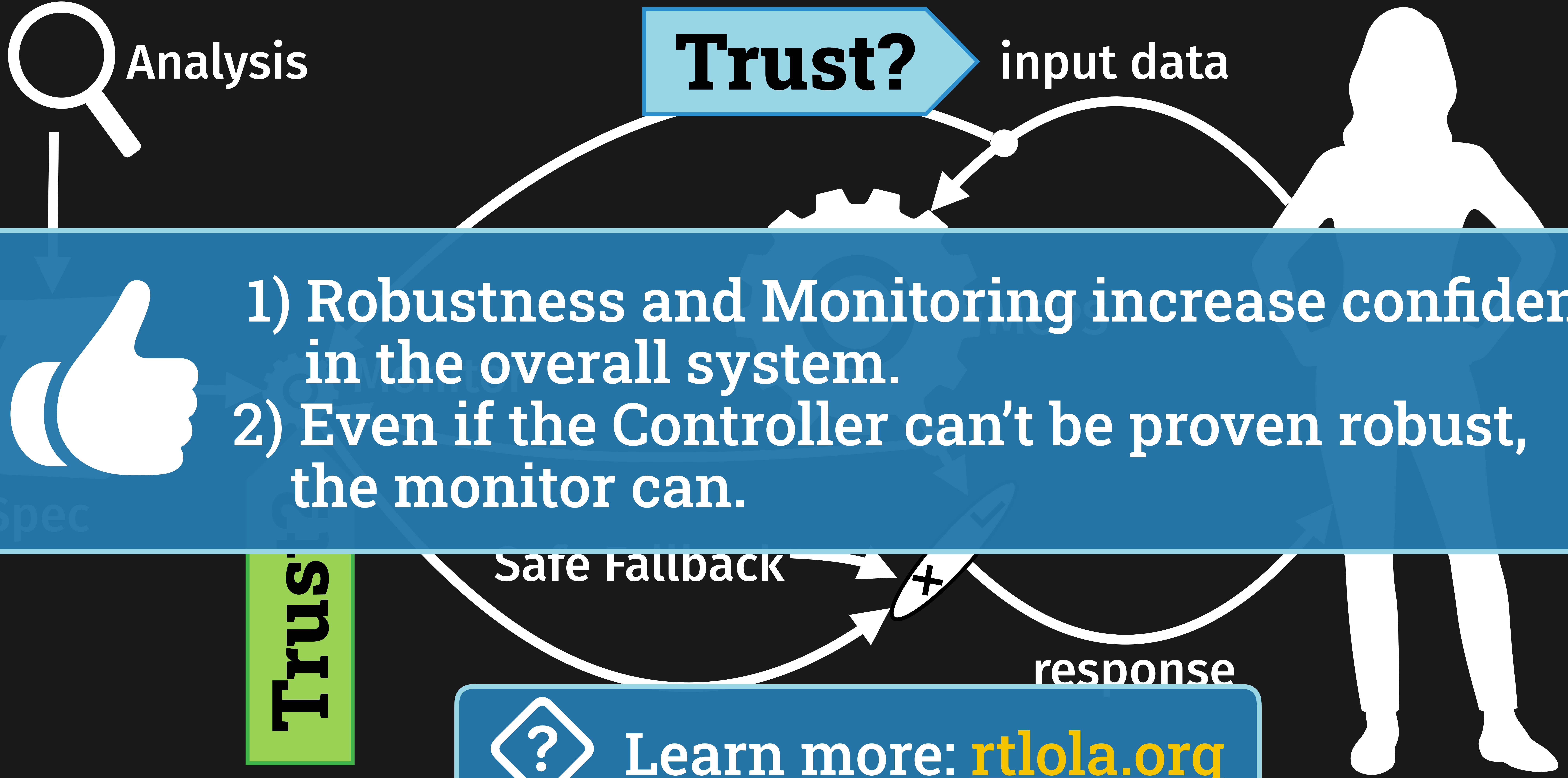

Analysis
Trust?
input data
MCPS
Monitor
Spec
Trust?
Safe Fallback
✓
×
response

Conclusion



1) Robustness and Monitoring increase confidence in the overall system.
2) Even if the Controller can’t be proven robust, the monitor can.

Conclusion



1) Robustness and Monitoring increase confidence in the overall system.
2) Even if the Controller can’t be proven robust, the monitor can.

Learn more: rtlola.org