Testing the Timing Robustness of the Functional Software Layer of an Autonomous Robot

David Powell, Hoang Nam Chu, Jean Arlat,
Félix Ingrand, Marc-Olivier Killijian

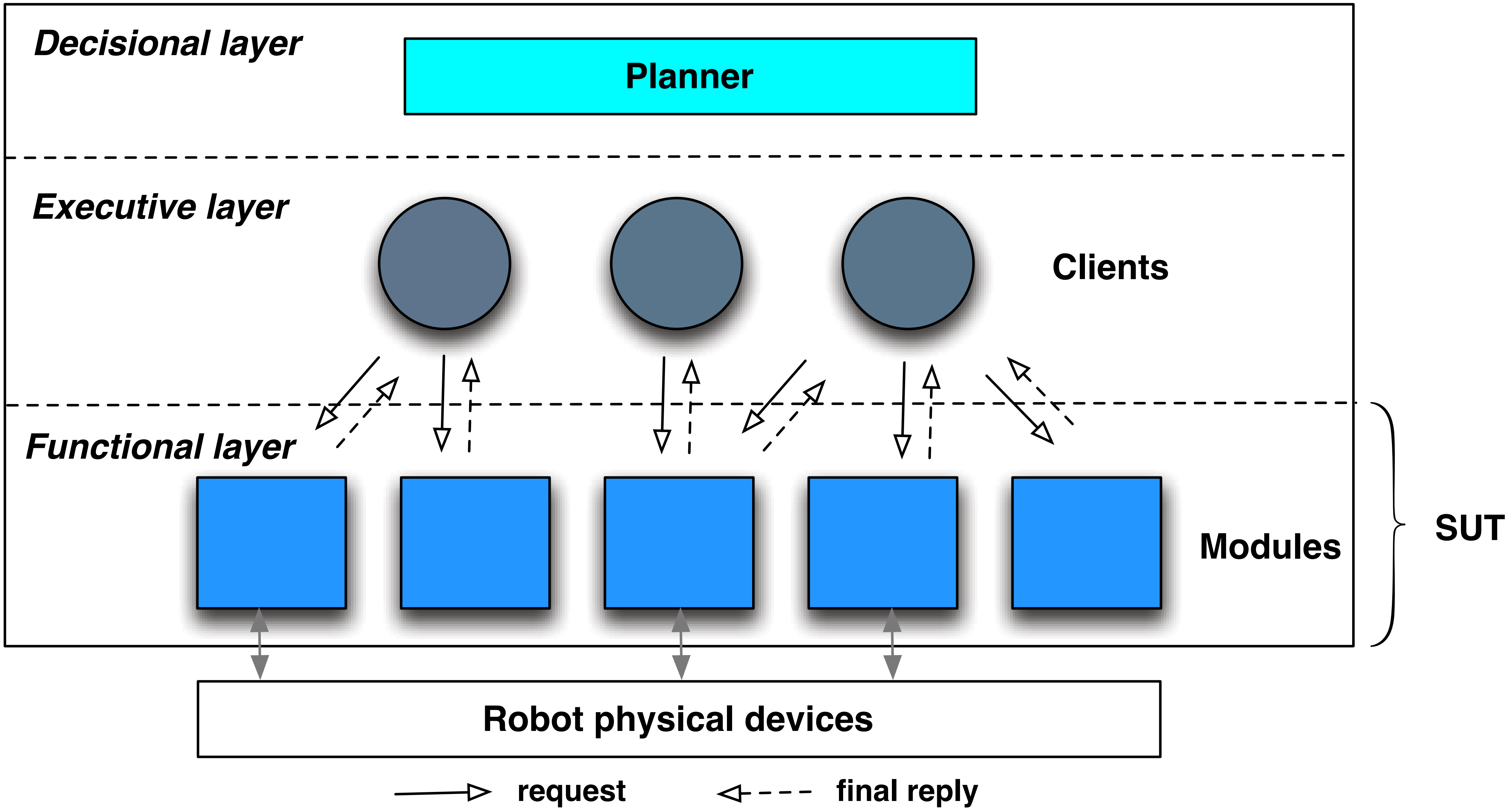
LAAS-CNRS

IFIP 10.4 Working Group Meeting, Saint-Luce, Martinique, 26-30 January 2012

Robustness

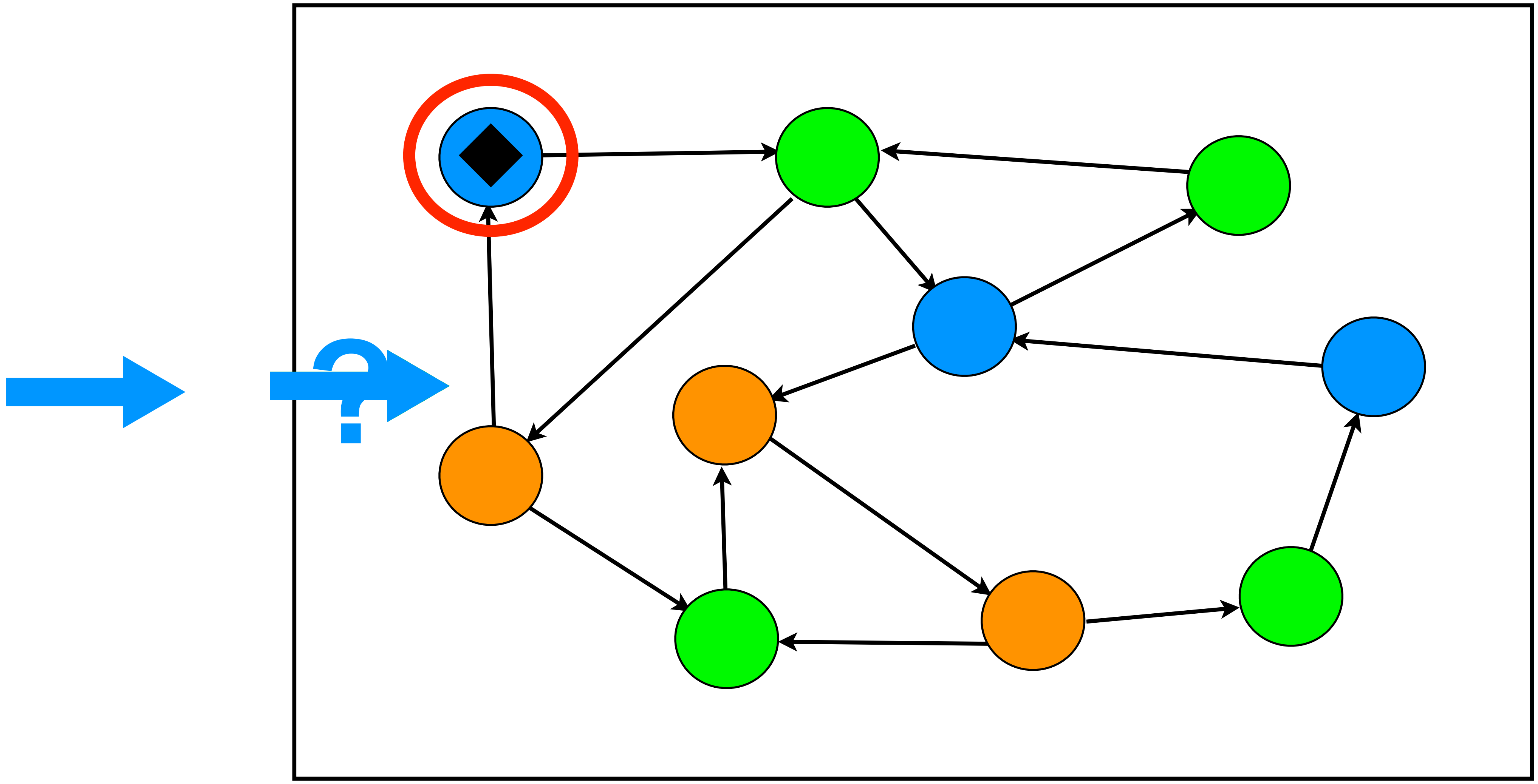
- **Degree to which system can function correctly in the presence of invalid inputs or stressful environmental conditions [IEEE Std. 610-12, 1990]**
  - **stressful environmental conditions**
    - non-functional stress (e.g., interference, temperature...)
    - functional stress (e.g., load, problem complexity...)
  - **invalid (functional) inputs**
    - invalid in value (e.g., requests with incorrect parameters)
    - invalid in time (e.g., requests at wrong time)

3-Layer Architecture



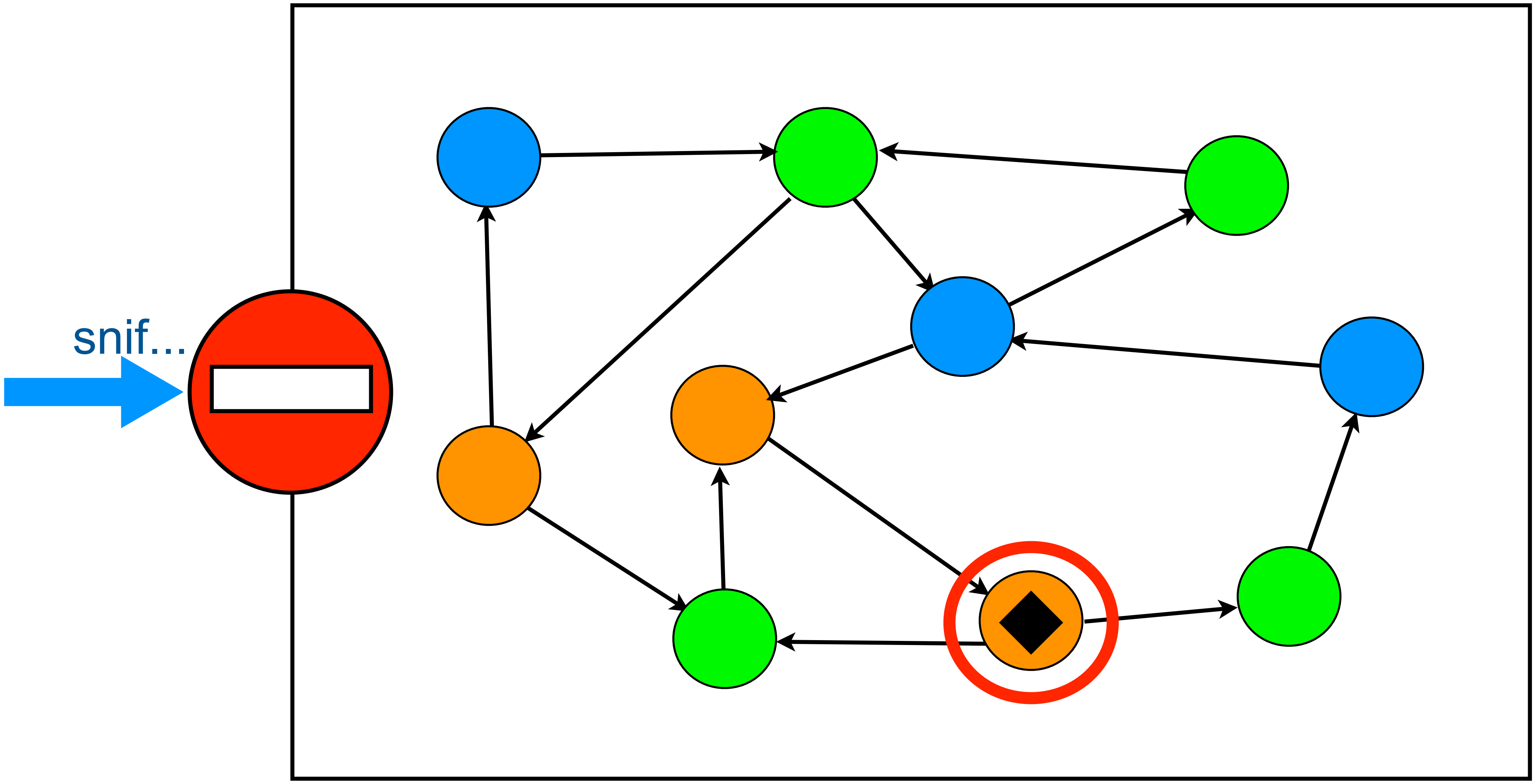
Input timing robustness

- Capacity of system to react to inputs that are sent at the "wrong" time



Input timing robustness

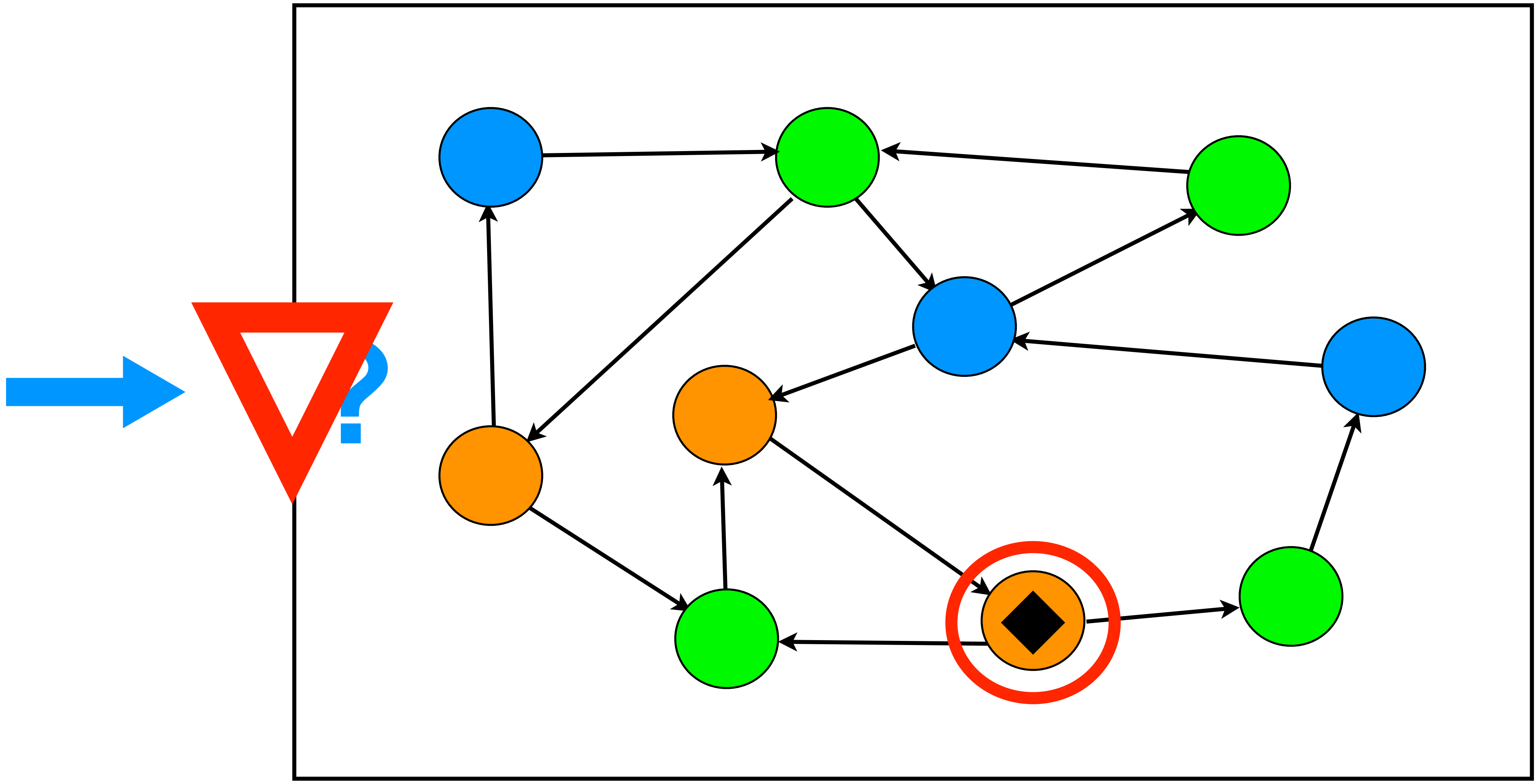
- Capacity of system to react to inputs that are sent at the "wrong" time



Reaction type 1 : reject input

Input timing robustness

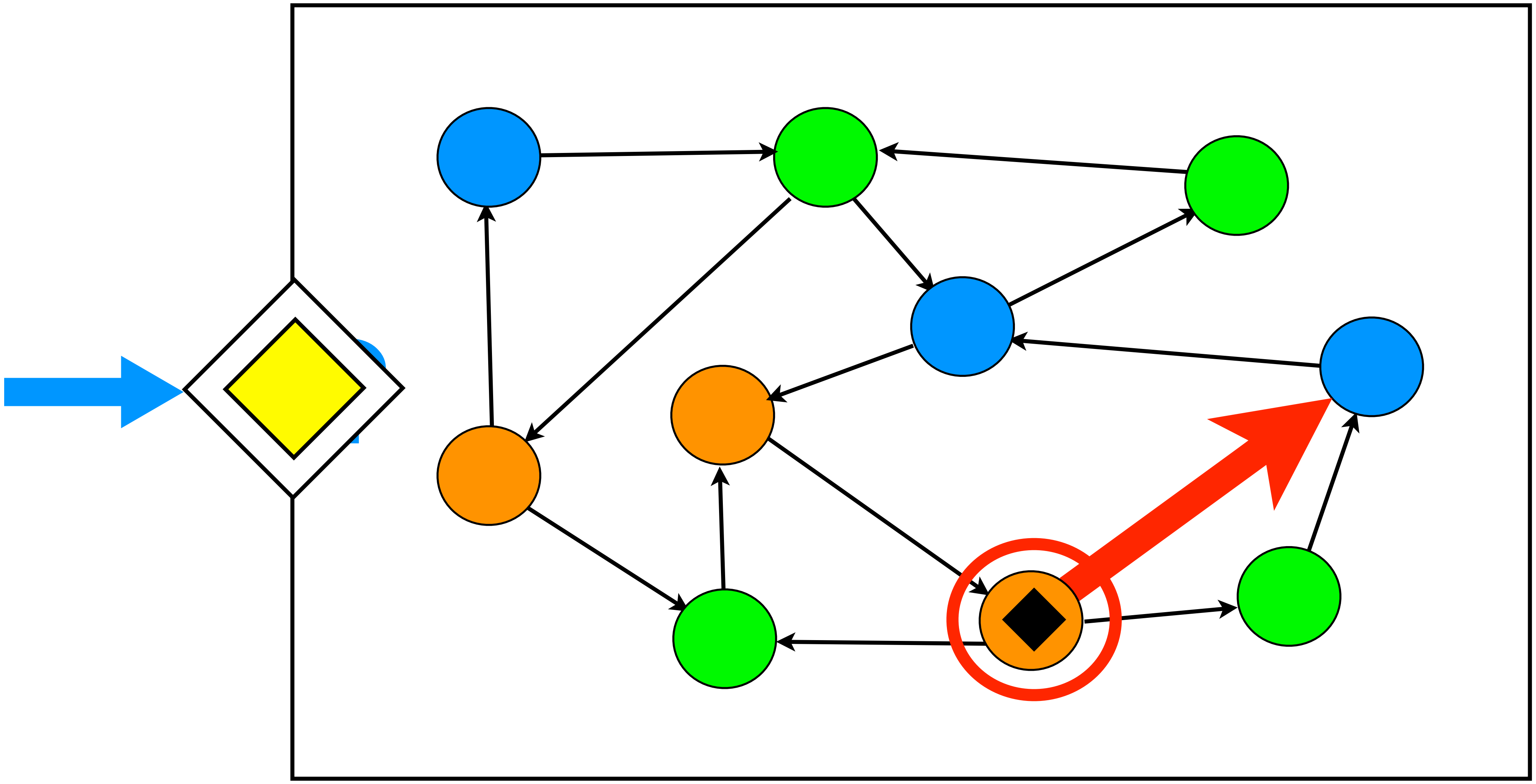
- Capacity of system to react to inputs that are sent at the "wrong" time



Reaction type 2 : queue input

Input timing robustness

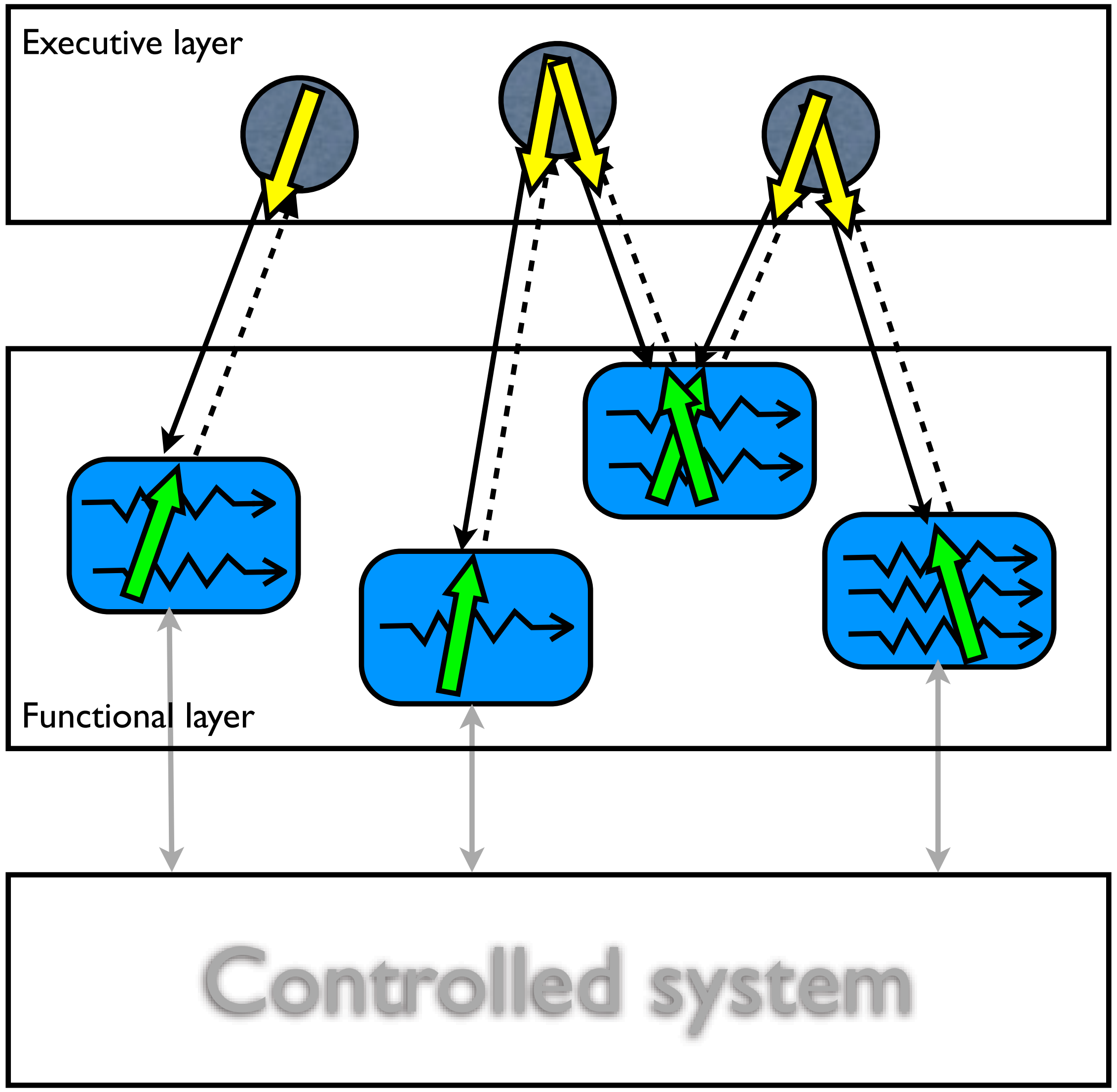
- Capacity of system to react to inputs that are sent at the "wrong" time



Reaction type 3 : force state change (e.g., interrupt)

Input timing robustness testing

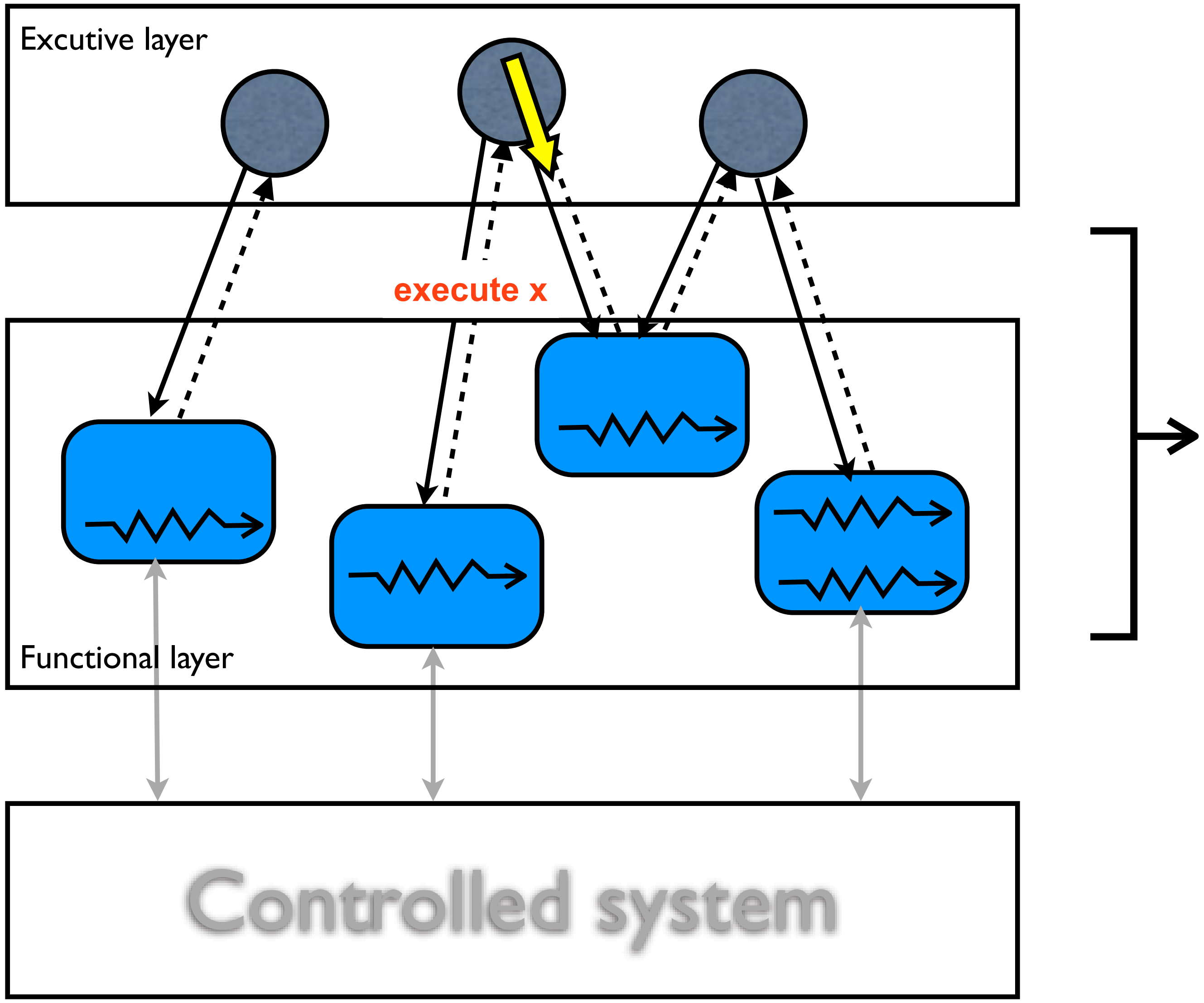
- Test capacity of system to react to inputs that are sent at the "wrong" time



Input timing robustness properties

- **Abstract state of the functional layer:**
  - initial state + history of activities executed to date
  - activites being executed now...

"Color"?



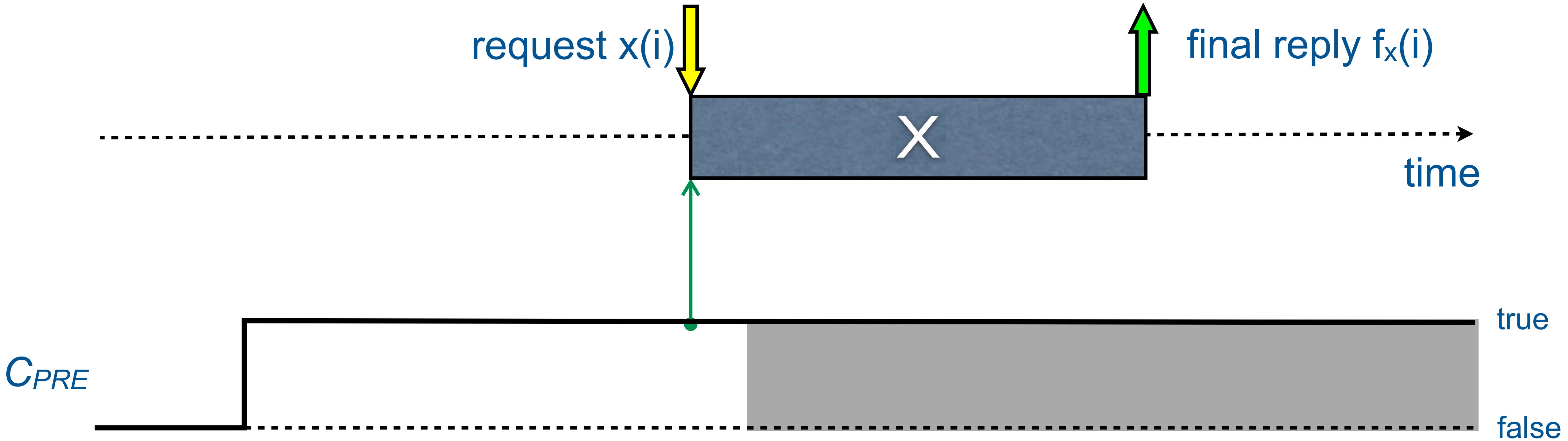
Set of properties

Input timing robustness properties

- **Precondition PC[x, C_{PRE}]**
  - start of activity x requires C_{PRE} = true
- **Exclusive start ES[x,y]**
  - activity x start excluded by ongoing activity y
- **Exclusive execution EE[x,y]**
  - activity x execution excluded by request for activity y
- **Exclusion EX[x,y]**
  - activity x excluded by activity y
    (EX[x,y] $\equiv$ ES[x,y] $\wedge$ EE[x,y])
- **Mutual exclusion MX[x,y]**
  - activities x and y cannot execute simultaneously

Robustness behavior categories

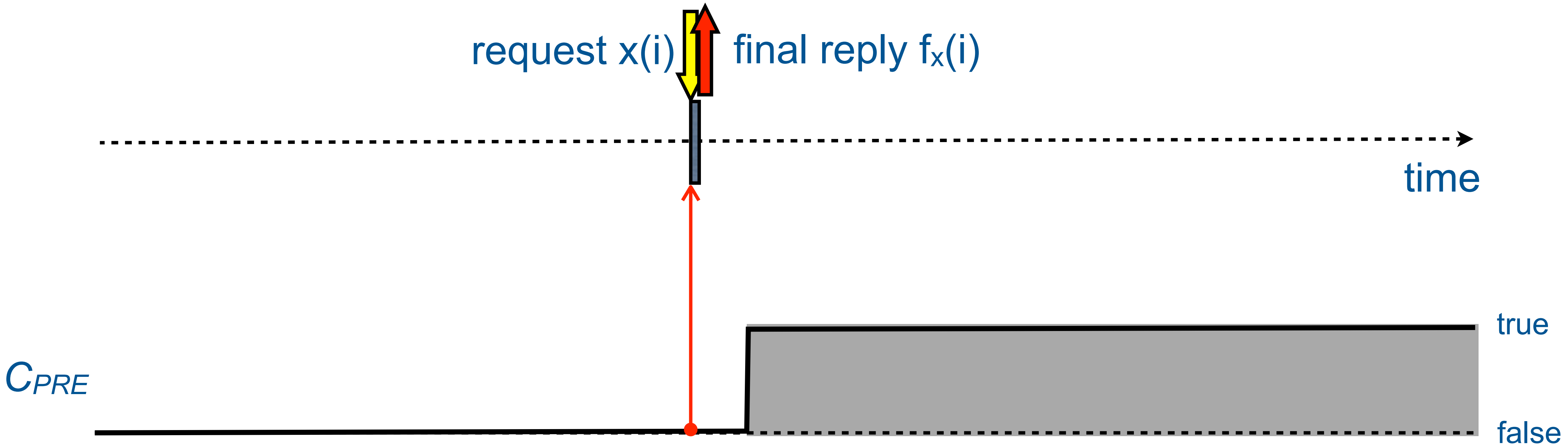
Precondition PC[x, C_{PRE}] enforced by rejection



True Negative (TN) : no invocation of property enforcement, since execution of request is authorized

Robustness behavior categories

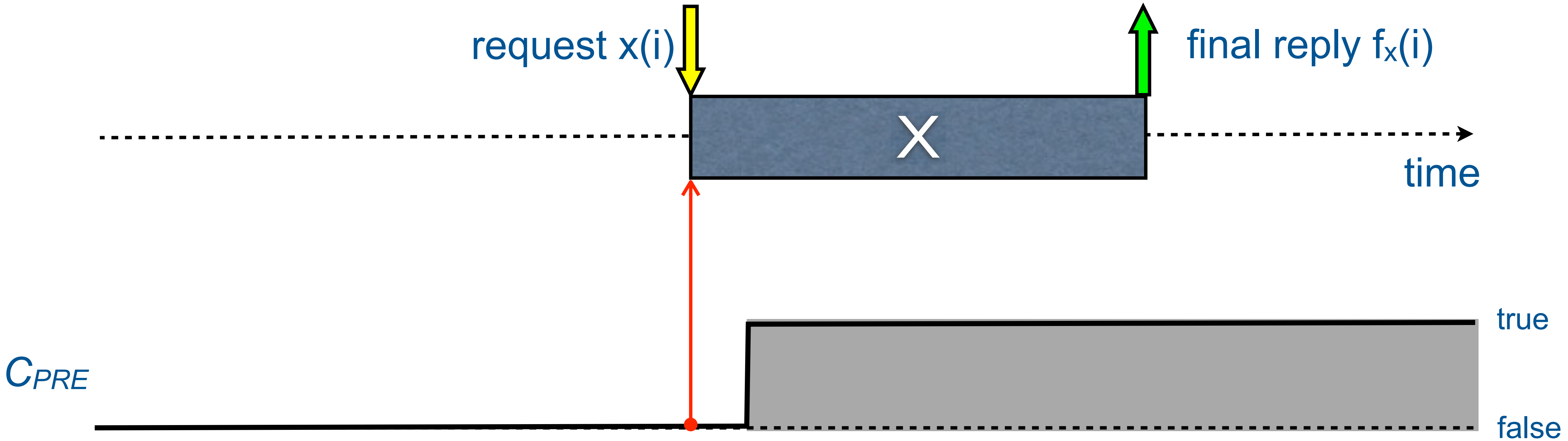
Precondition PC[x, C_{PRE}] enforced by rejection



True Positive (TP) : invocation of property enforcement (rejection), since execution of request is *not authorized*

Robustness behavior categories

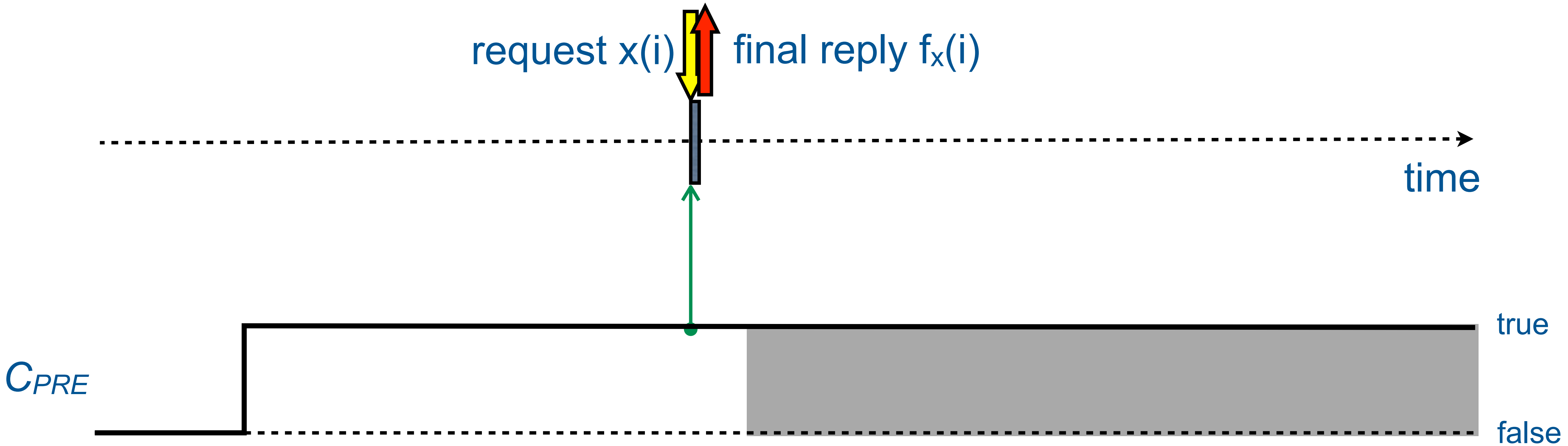
Precondition PC[x, C_{PRE}] enforced by rejection



False Negative (FN) : no invocation of property enforcement, yet execution of request is *not authorized*

Robustness behavior categories

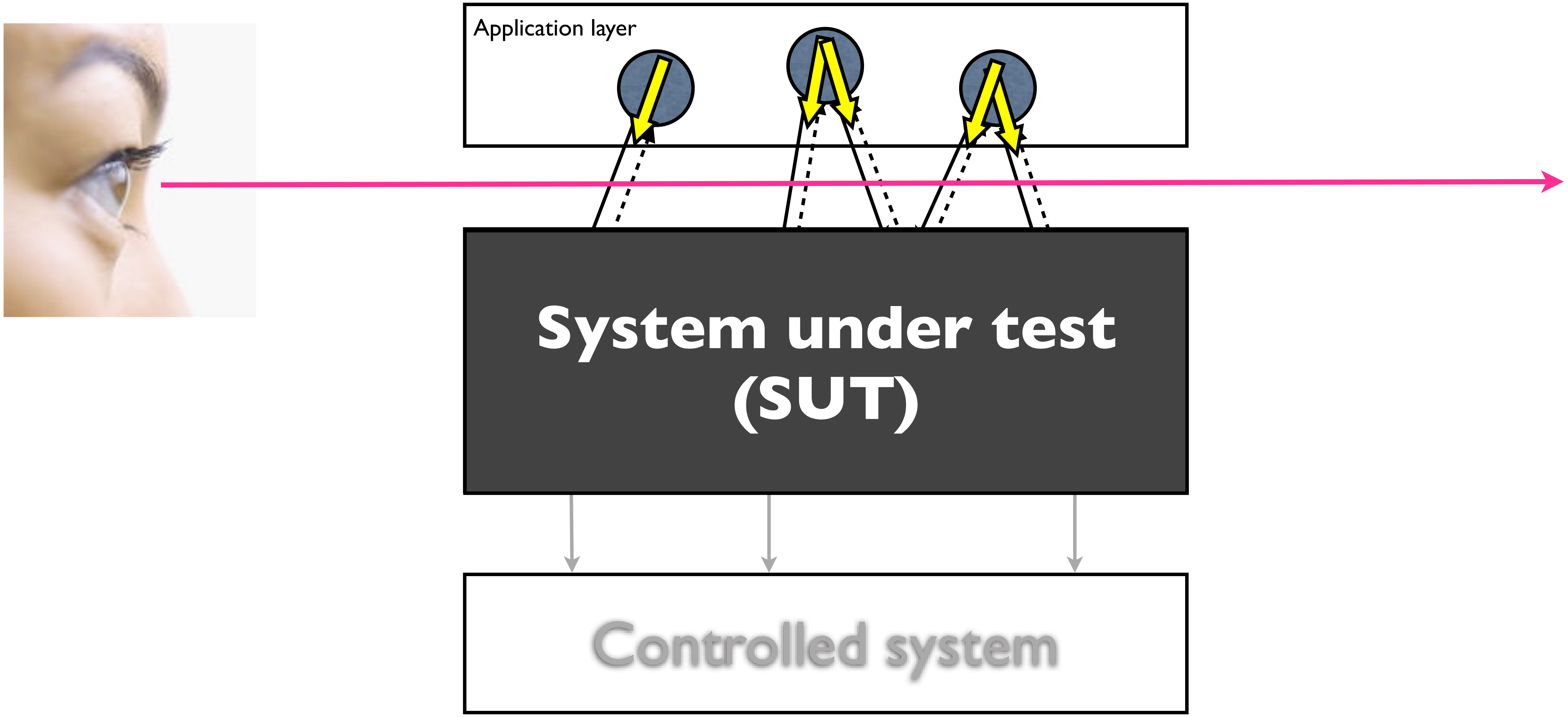
Precondition PC[x, C_{PRE}] enforced by rejection



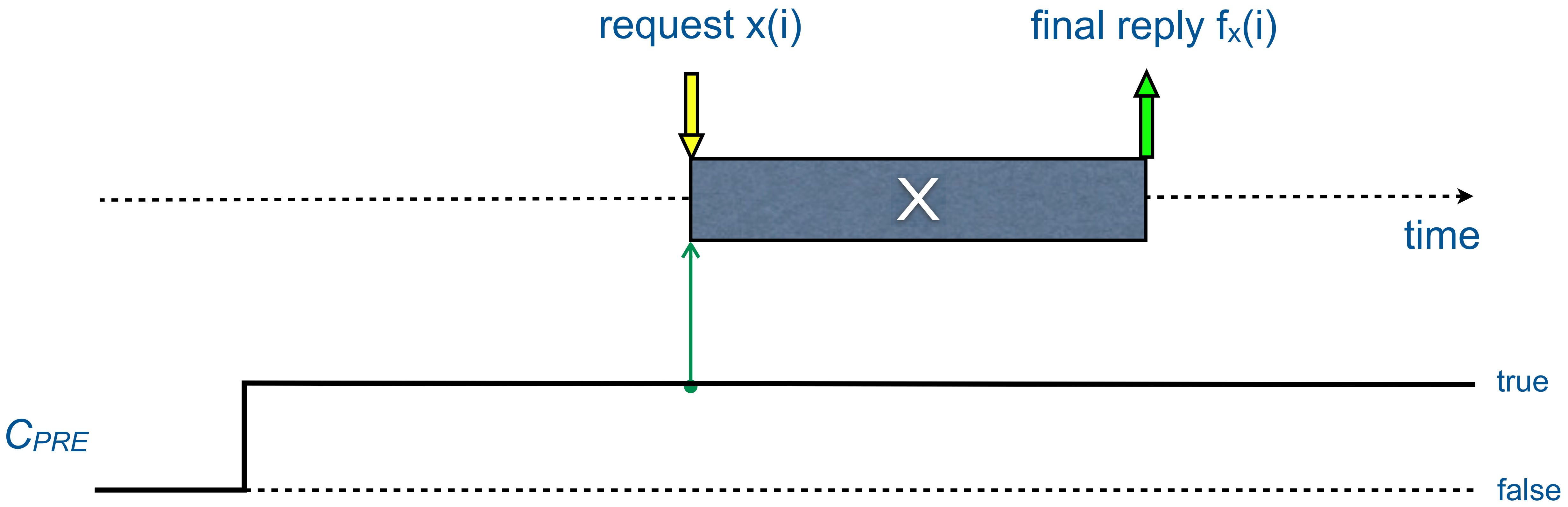
False Positive (FP) : invocation of property enforcement (rejection), but execution of request is *authorized*

Black-box robustness testing

- Testing at the interface, with no access to internals of system under test
  - ☺ Enables comparison of different SUTs (with same interface)

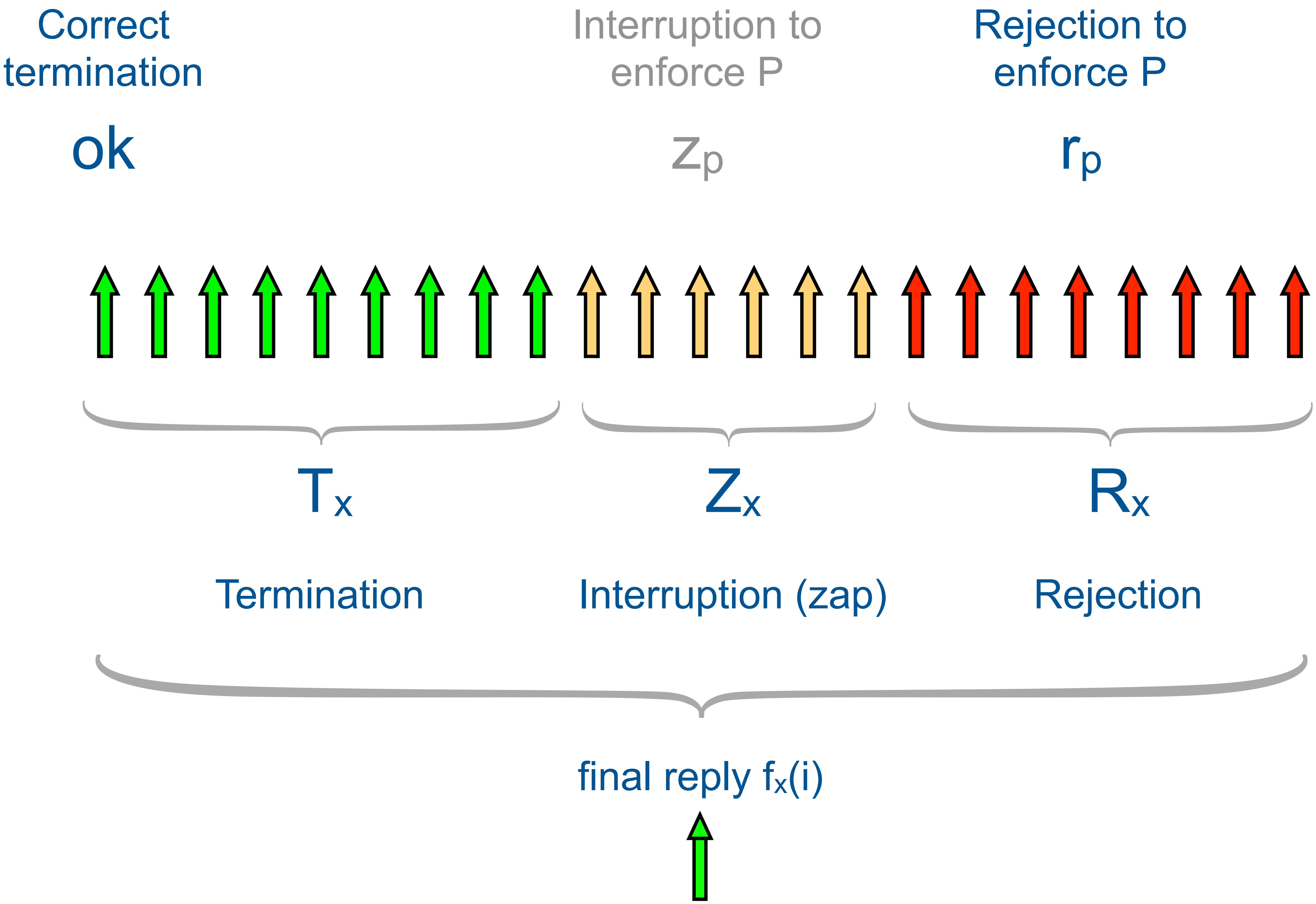


Oracle for property PC[x,C_{PRE}]



Oracle for property PC[x,C$_{PRE}$]

Some notation



Correct termination

ok

Interruption to enforce P

z_p

Rejection to enforce P

r_p

T_x

Termination

Z_x

Interruption (zap)

R_x

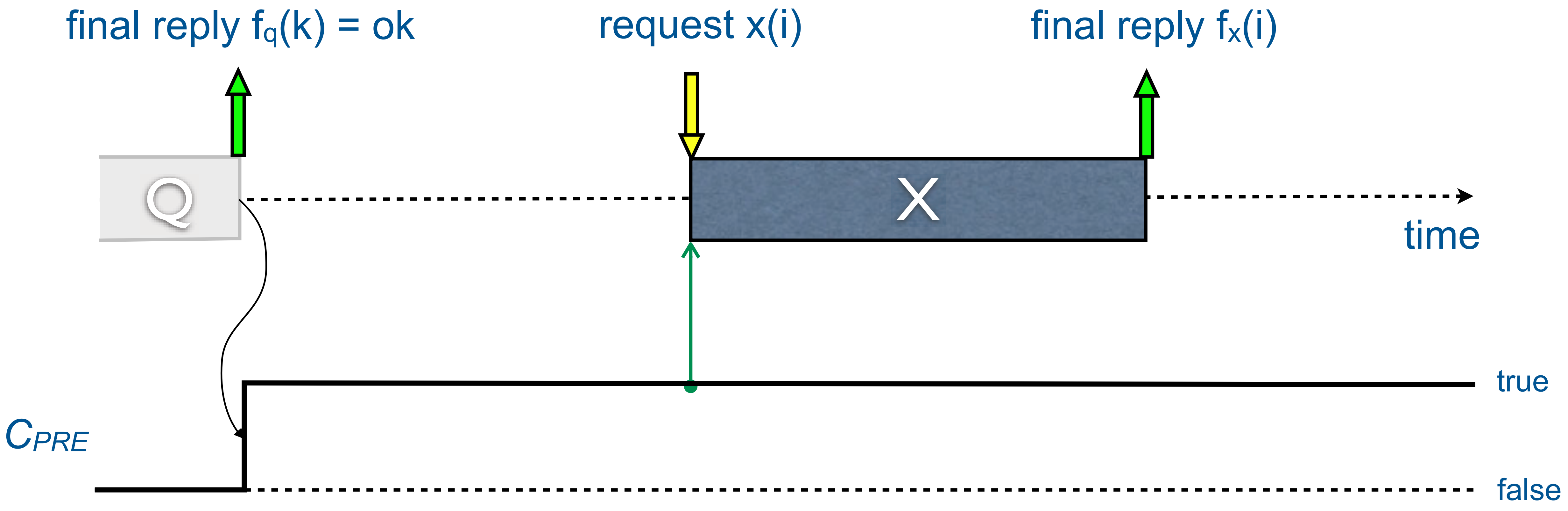
Rejection

final reply $f_x(i)$

Oracle for property $PC[x, C_{PRE}]$

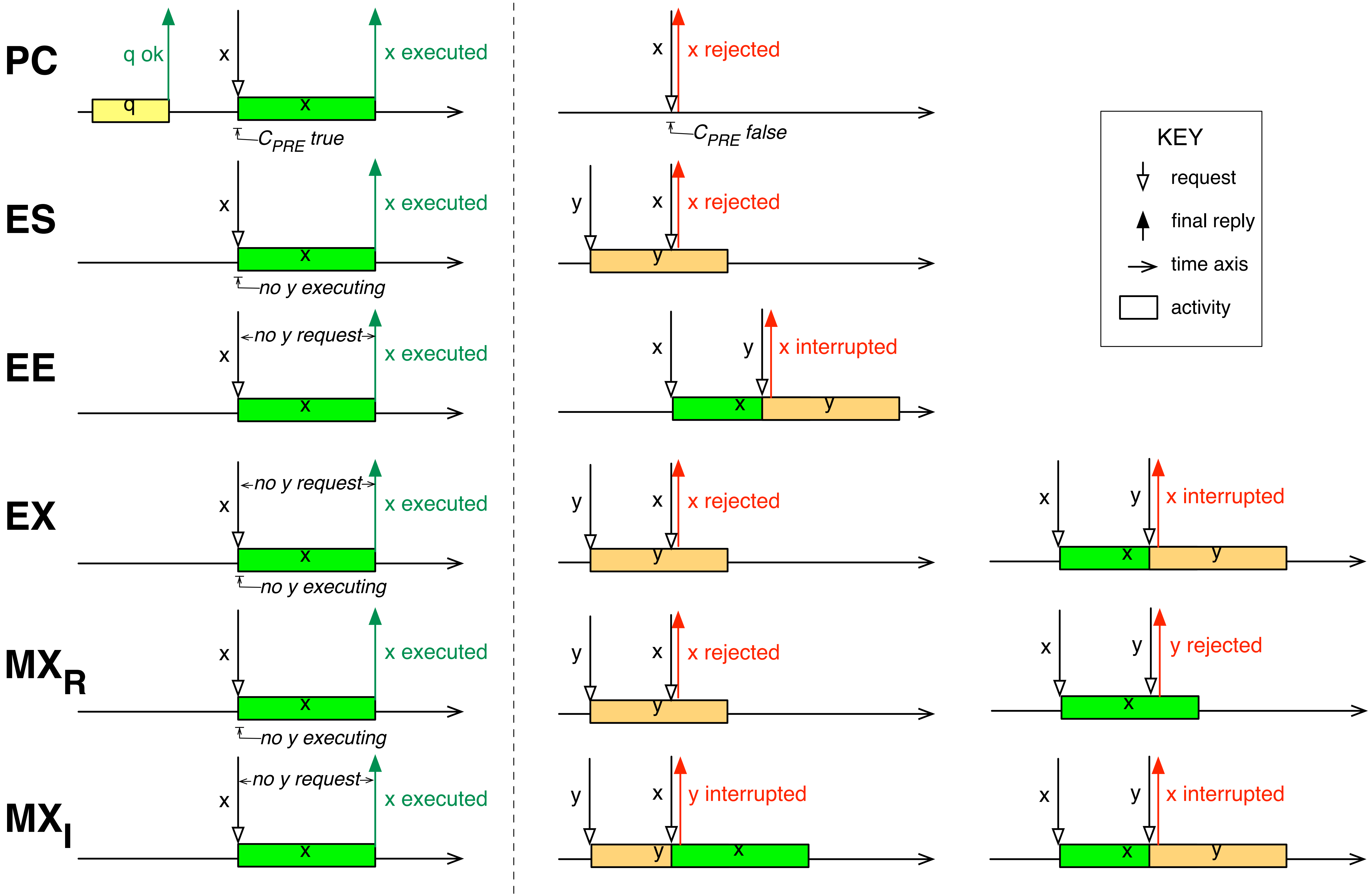
Example: C_{PRE} = activity Q successfully completed

$$C_{PRE}(x(i)) = \exists\, f_q(k),\ [t(f_q(k)) < t(x(i))] \wedge [f_q(k) = ok]$$



		$f_x(i)$			
		$\in \{Z_x, T_x\}$	r_P	$\in R_x \setminus r_P$	ø
$C_{PRE}(x(i))$	true	TN	FP	**OP** (other positive)	w
	false	FN	TP		

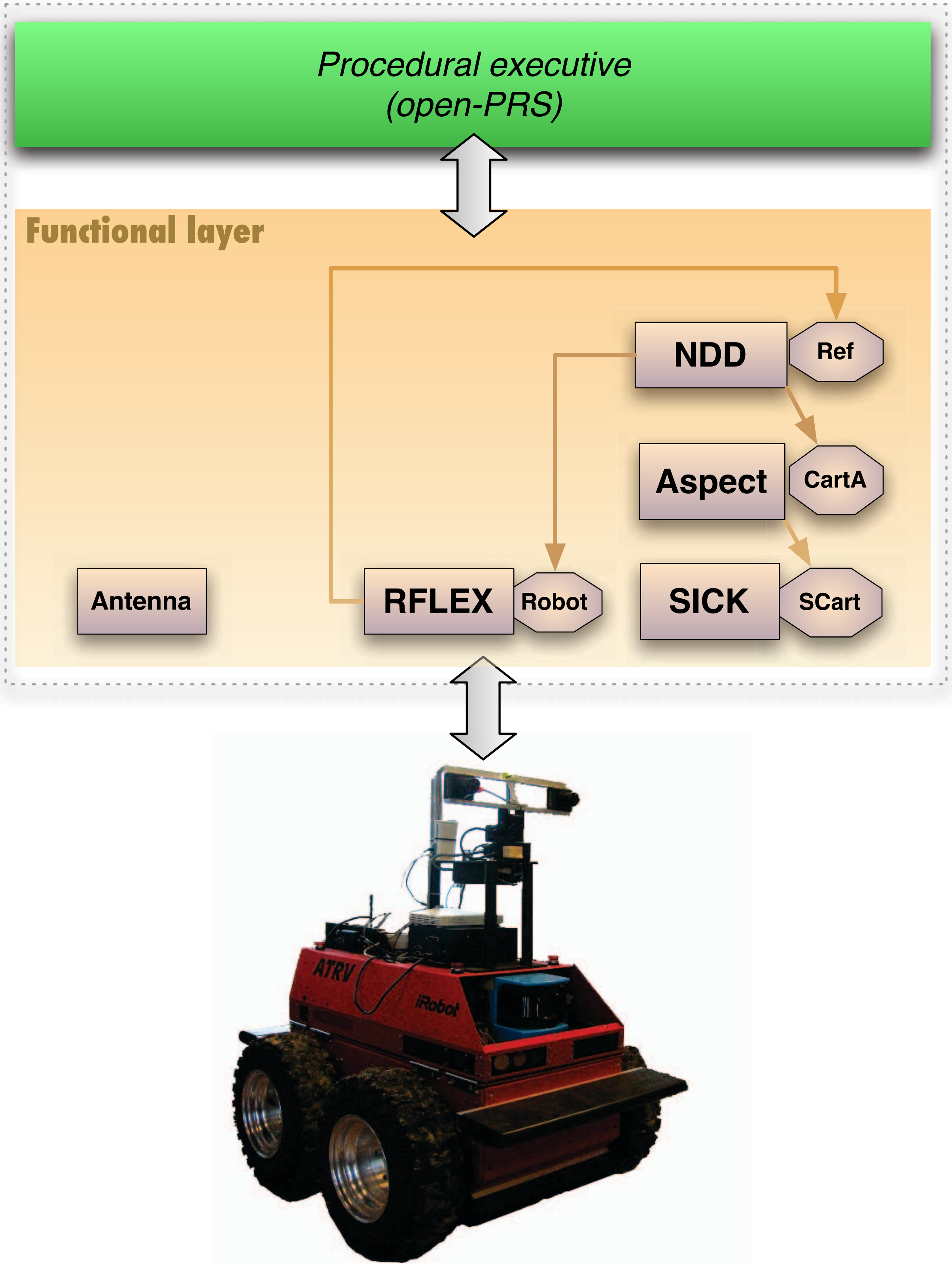
Oracle types

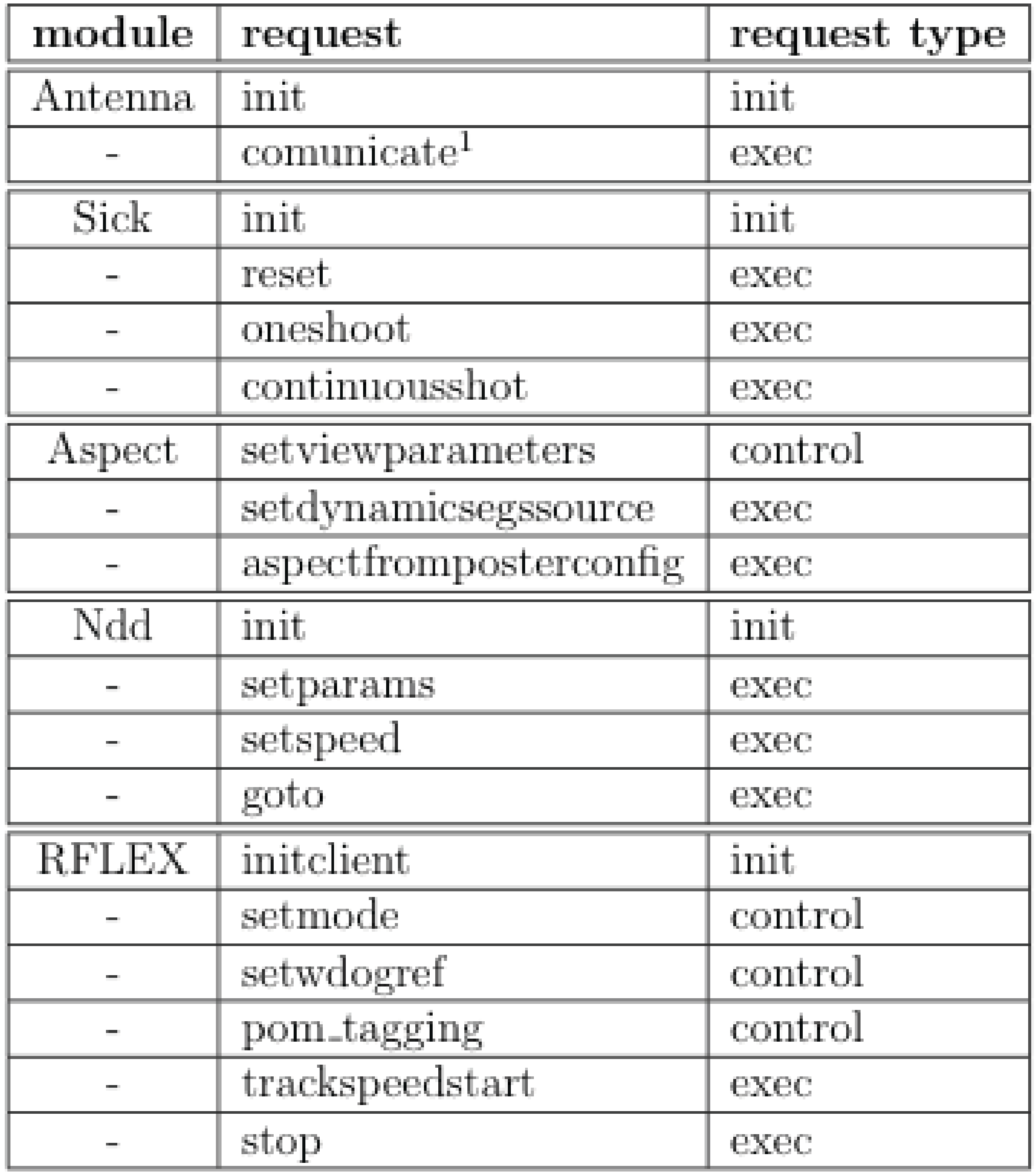


True negative behaviors

True positive behaviors

Case study: the Dala rover



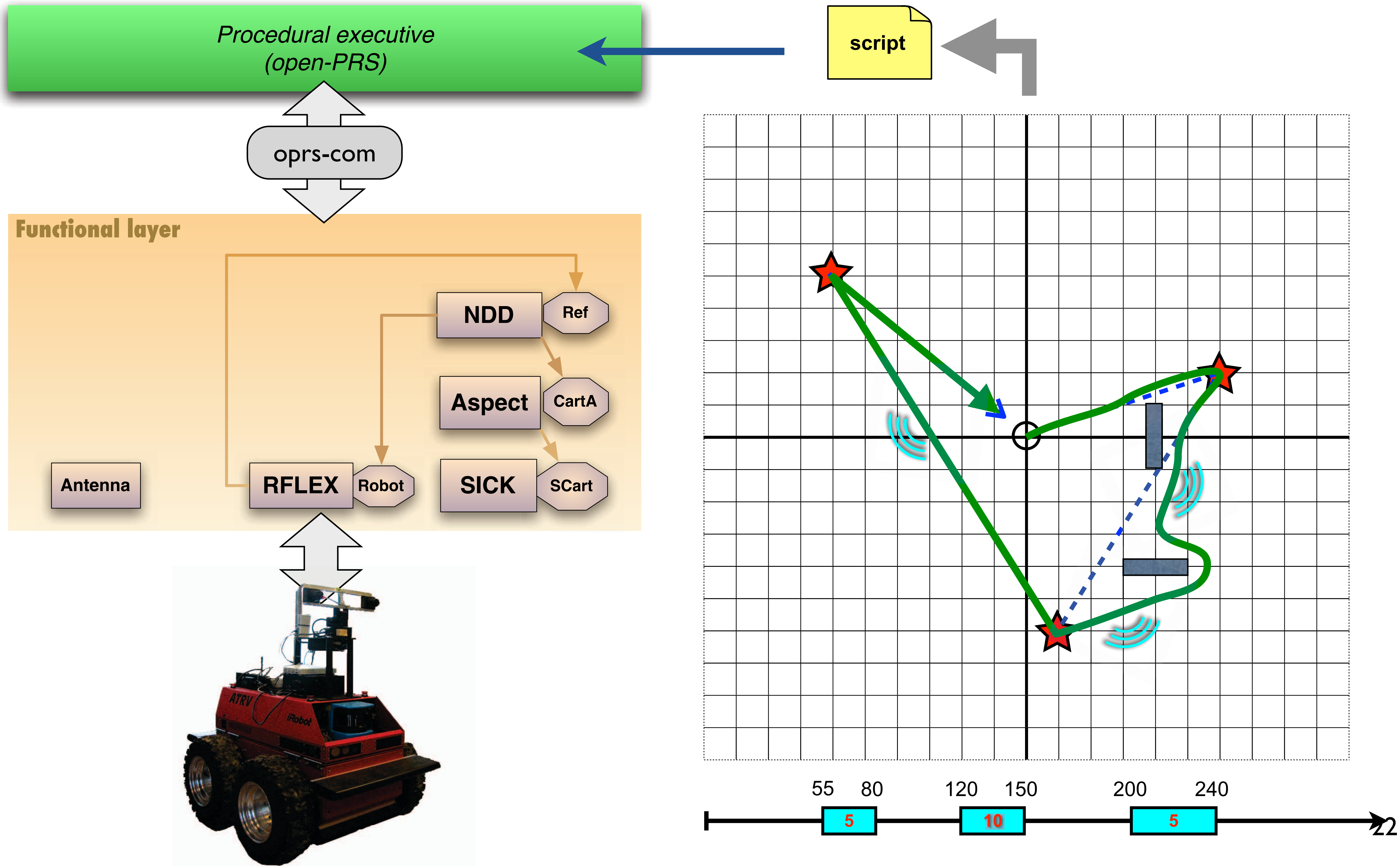


module	request	request type
Antenna	init	init
-	comunicate[1]	exec
Sick	init	init
-	reset	exec
-	oneshoot	exec
-	continuousshot	exec
Aspect	setviewparameters	control
-	setdynamicsegssource	exec
-	aspectfromposterconfig	exec
Ndd	init	init
-	setparams	exec
-	setspeed	exec
-	goto	exec
RFLEX	initclient	init
-	setmode	control
-	setwdogref	control
-	pom_tagging	control
-	trackspeedstart	exec
-	stop	exec

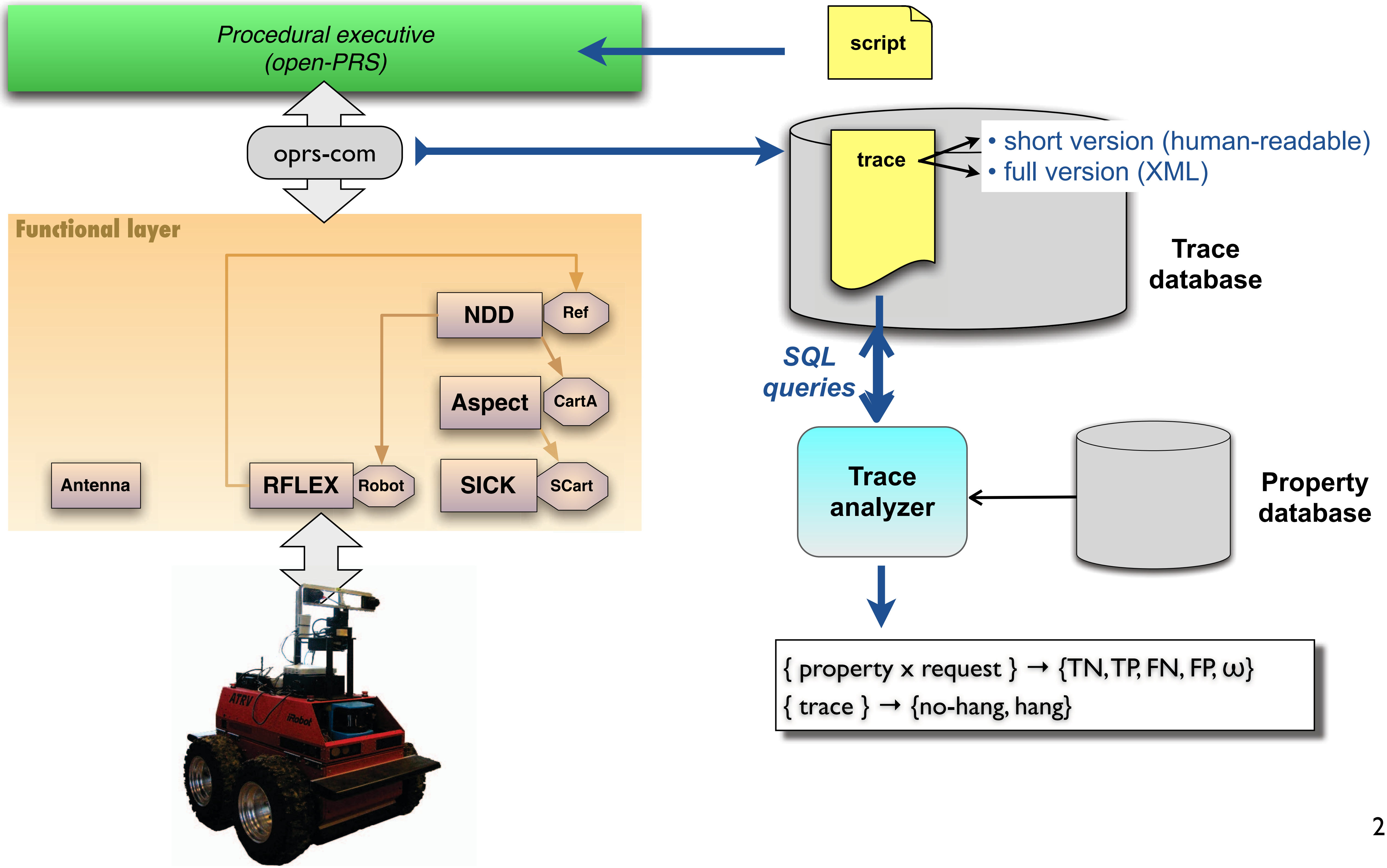
Dala properties

Property family	Definition	Modules					Oracle type
		Antenna	Sick	Aspect	Ndd	Rflex	
PEX	initialization must **P**recede **EX**ec requests (4 instances)	1	1		1	1	*PC*
AIB	**A**ctivity *x* **I**nterrupted **B**y *Y* (15 instances)	2	4	2	4	3	*EE*
PRE	aspect.setviewparameter & aspect.setdynamicsource must **PRE**cede aspect.aspectfromposterconfig			1			*PC*
	ndd.setparams & ndd.setspeed must **PRE**cede ndd.goto				1		
EXC	antenna.communicate & rflex.trackspeedstart are mutually EXClusive	1 ○—	—	—	—	—○	*MX*

Test environment



Test environment



Procedural executive
(open-PRS)

script

oprs-com

trace

- short version (human-readable)
- full version (XML)

Trace database

Functional layer

NDD

Ref

Aspect

CartA

Antenna

RFLEX

Robot

SICK

SCart

SQL queries

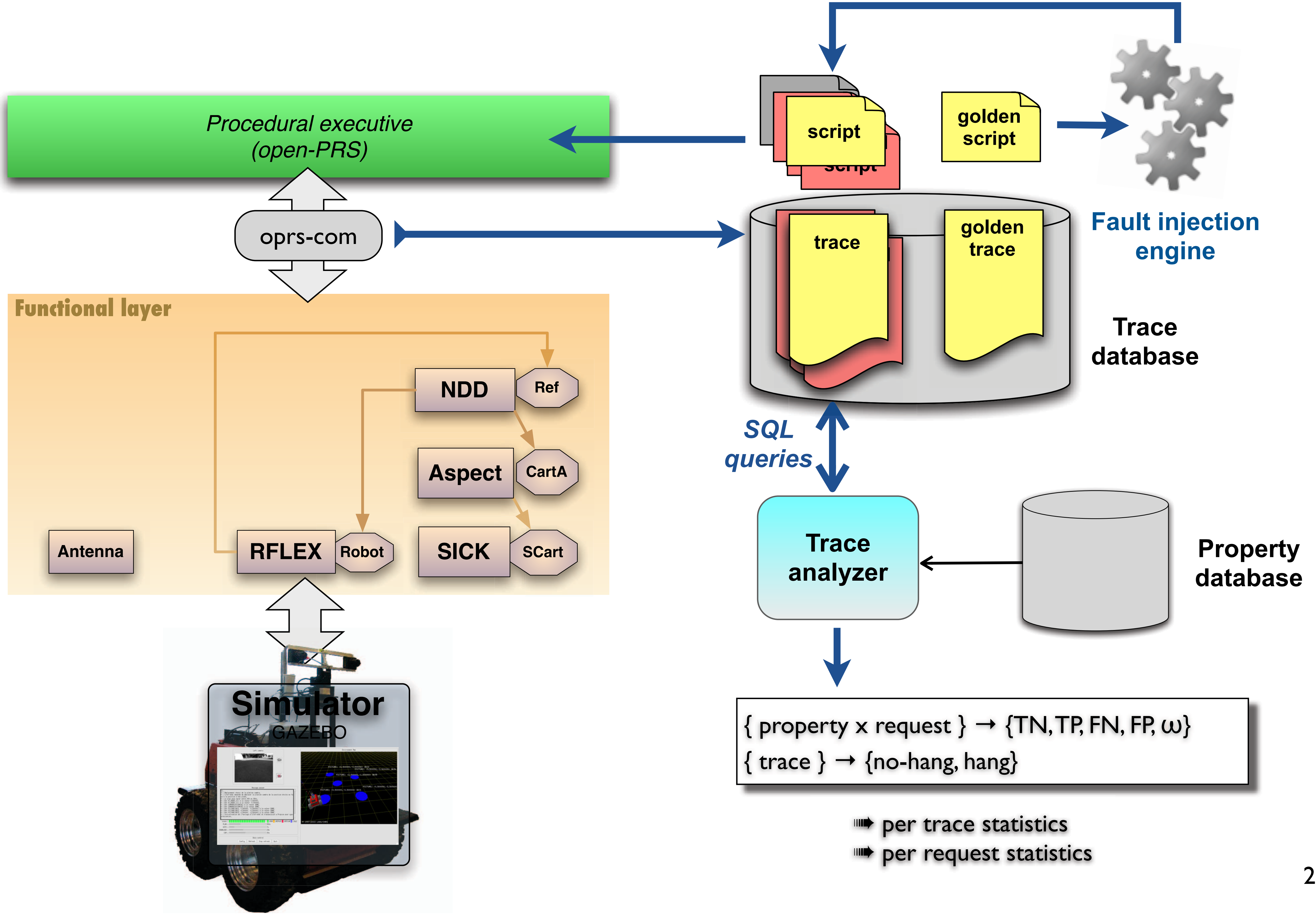
Trace analyzer

Property database

{ property x request } → {TN, TP, FN, FP, ω}

{ trace } → {no-hang, hang}

Test environment



Procedural executive
(open-PRS)

oprs-com

Functional layer

NDD

Ref

Aspect

CartA

Antenna

RFLEX

Robot

SICK

SCart

Simulator
GAZEBO

script

script

golden
script

Fault injection
engine

trace

golden
trace

Trace
database

SQL
queries

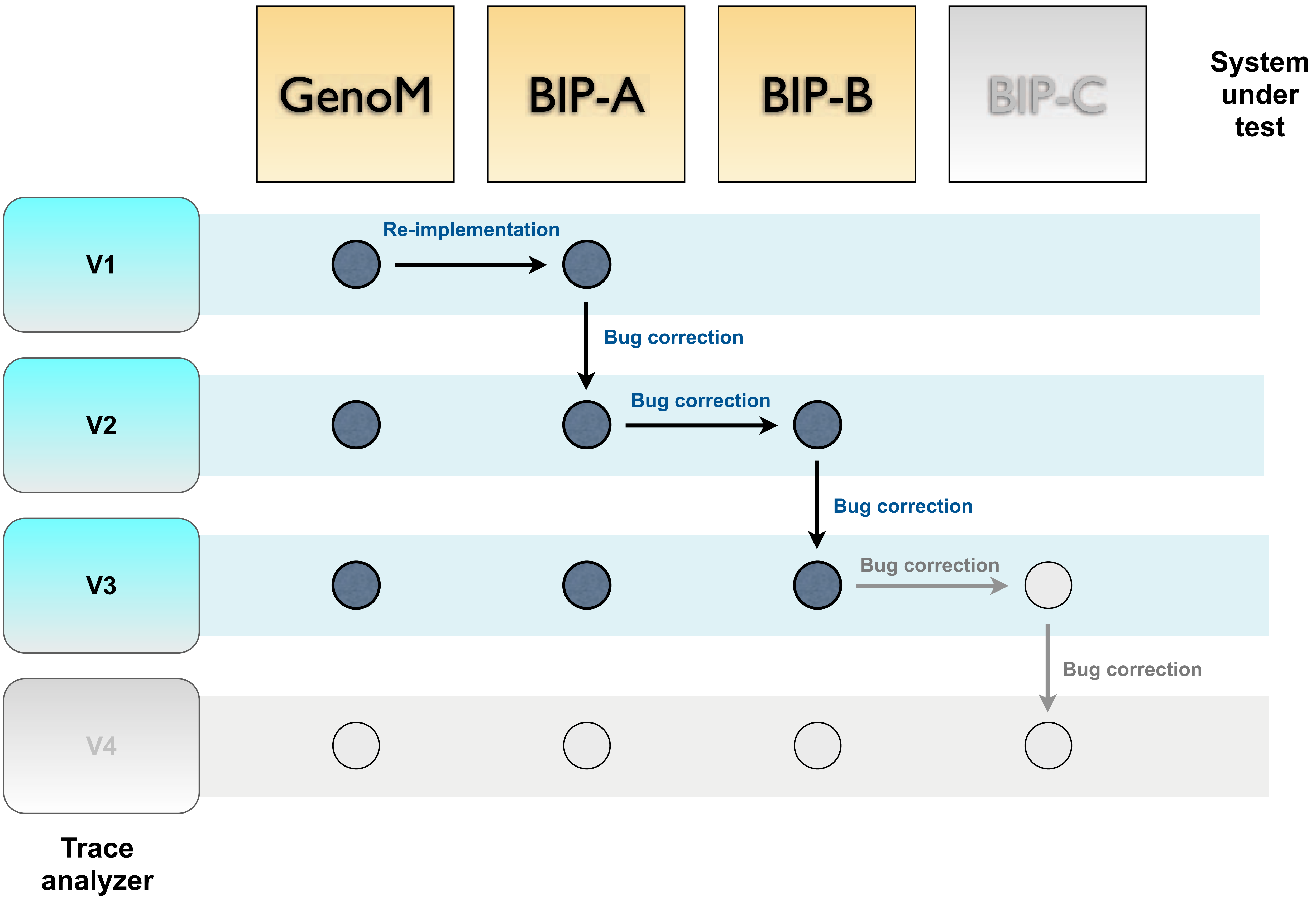
Trace
analyzer

Property
database

{ property x request } → {TN, TP, FN, FP, ω}
{ trace } → {no-hang, hang}

- per trace statistics
- per request statistics

Versions



GenoM

BIP-A

BIP-B

BIP-C

System
under
test

V1

Re-implementation

Bug correction

V2

Bug correction

Bug correction

V3

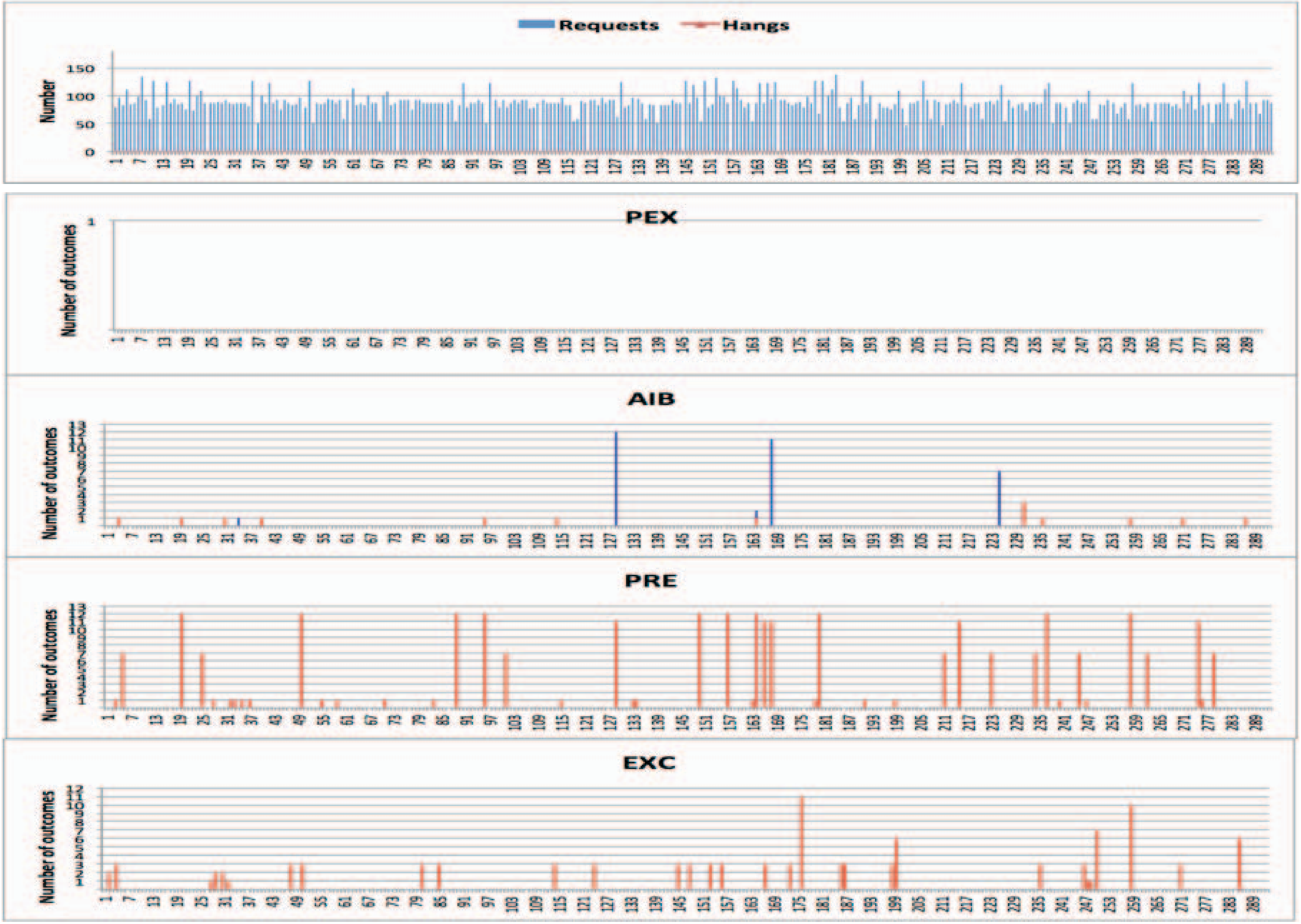
Bug correction

Bug correction

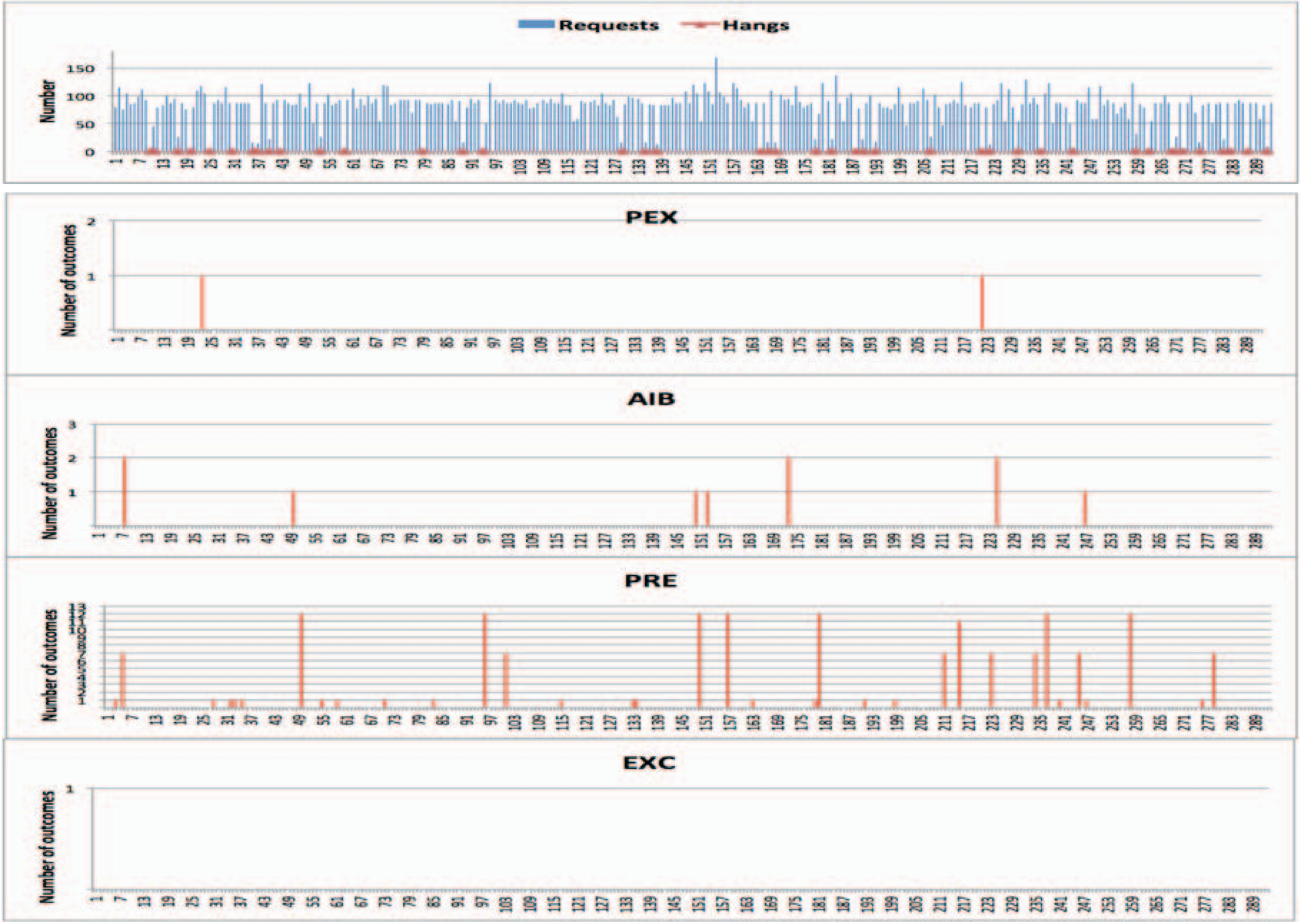
V4

Trace
analyzer

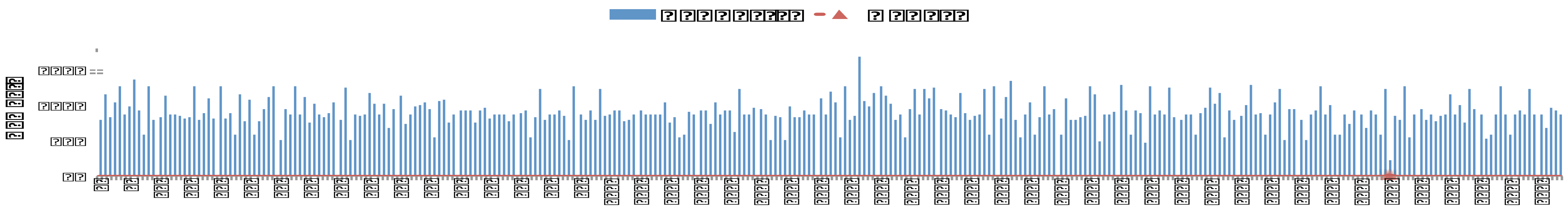
GenoM implementation

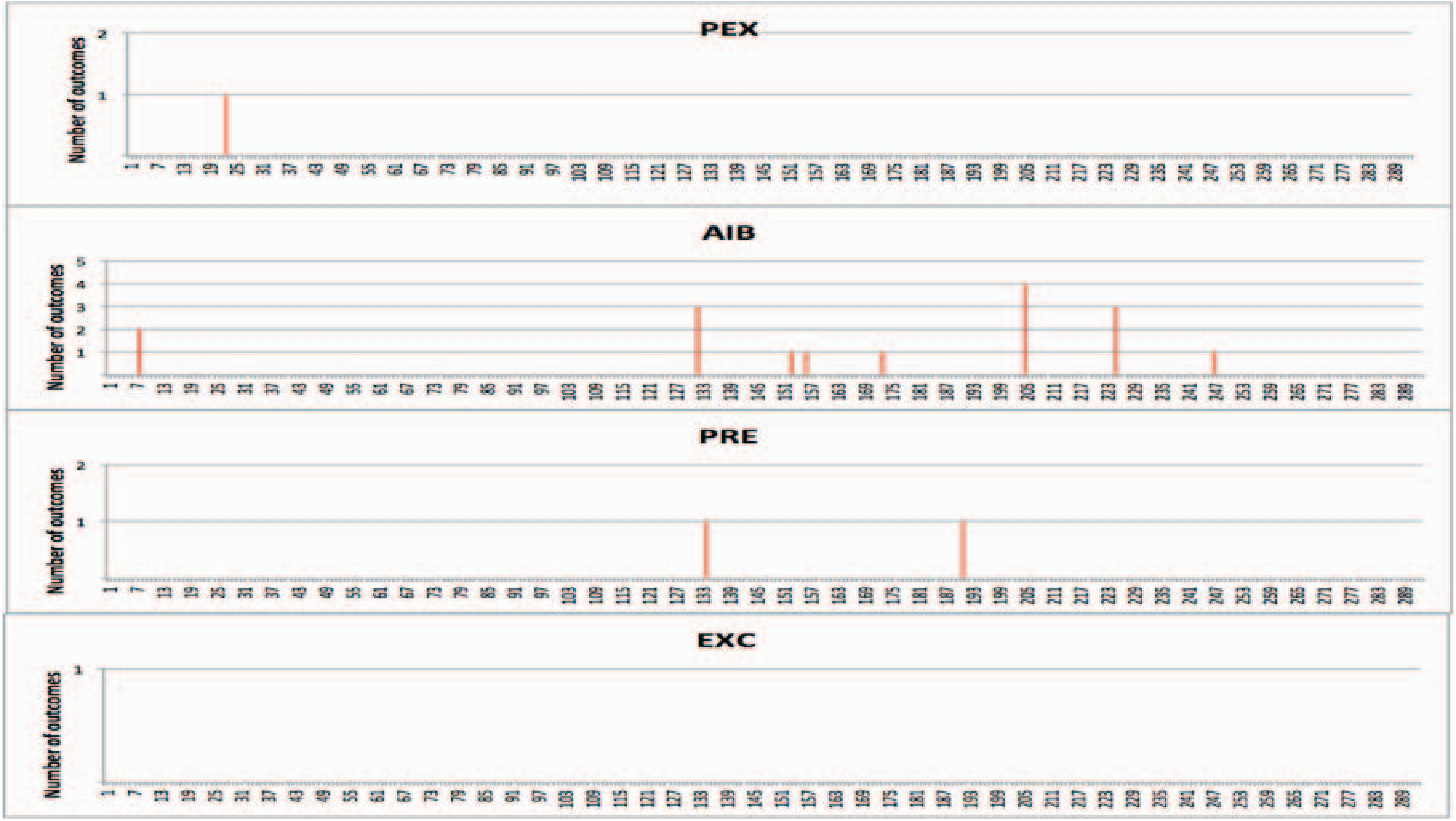


BIP implementation A



BIP implementation B

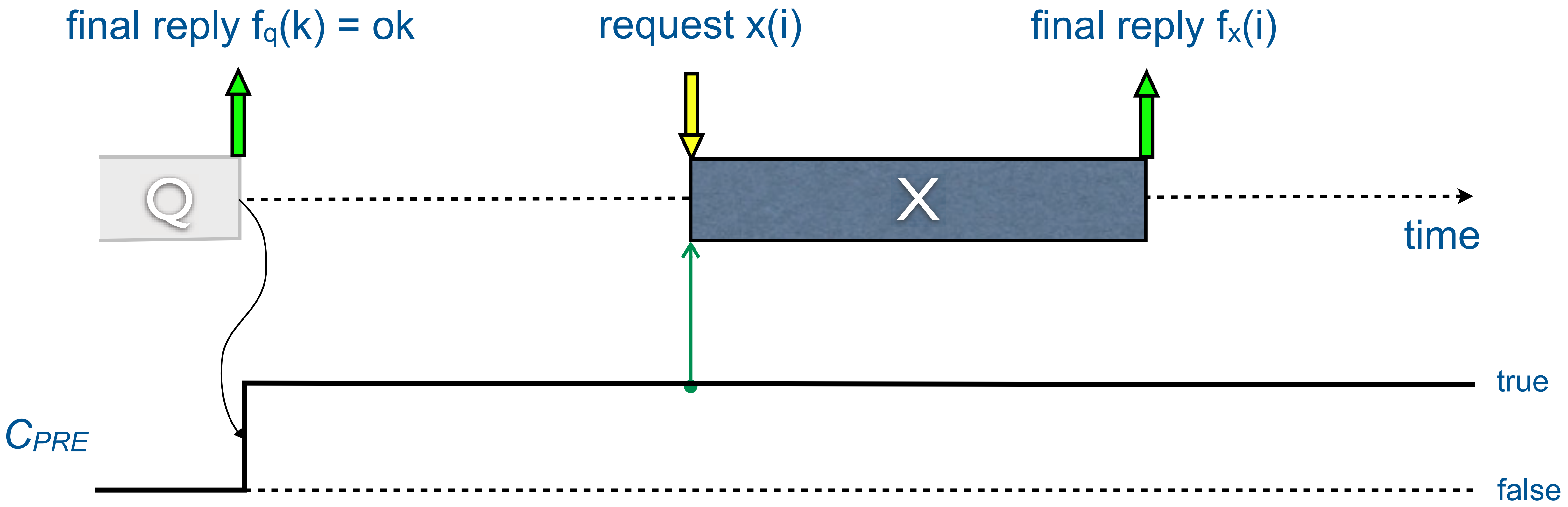




Oracle for property PC[x,C$_{PRE}$]

Example: C_{PRE} = activity Q successfully completed

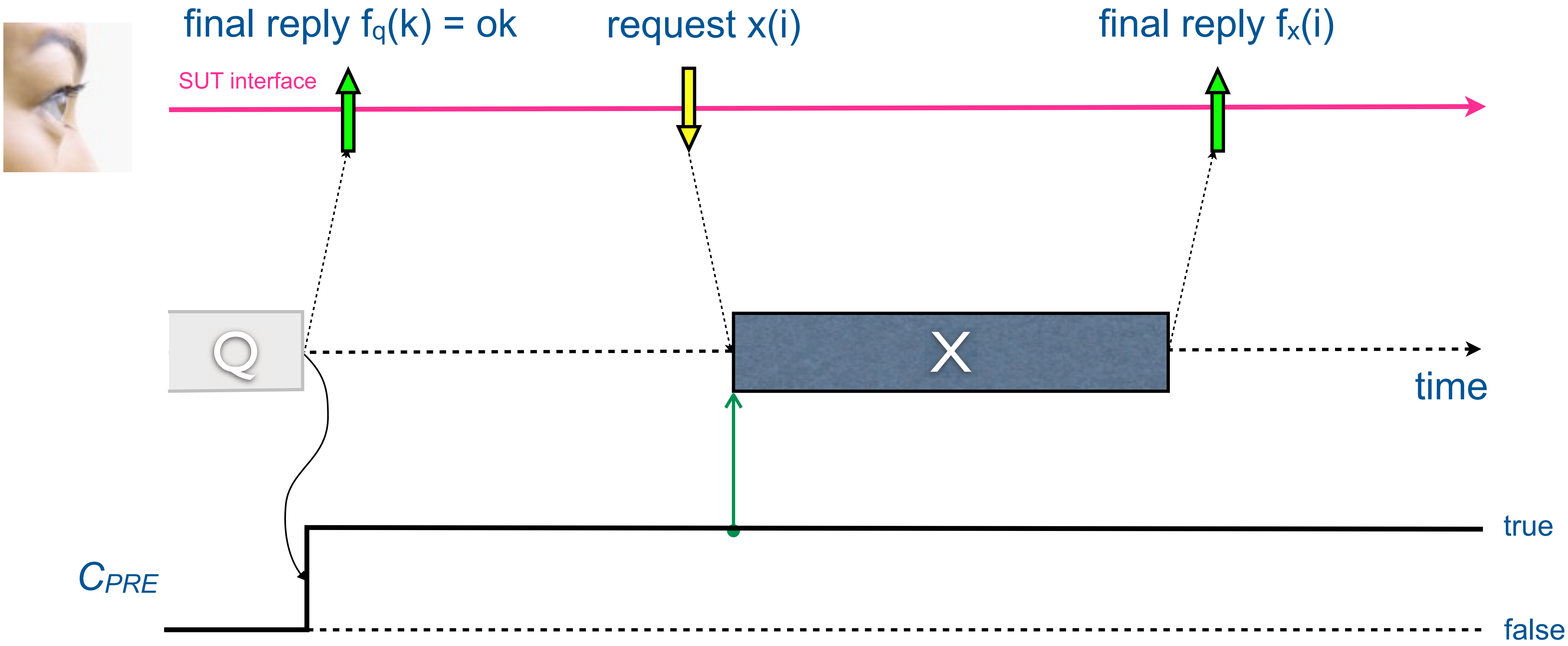
$$C_{PRE}\,(x(i)) = \exists\ f_q(k),\ [t(f_q(k)) < t(x(i))] \wedge [f_q(k) = ok]$$



		$f_x(i)$		
		$\in \{Z_x, T_x\}$	r_P	$\in R_x \setminus r_P$
$C_{PRE}\,(x(i))$	true	**TN**	**FP**	**OP** (other positive)
	false	**FN**	**TP**	

Oracle for property PC[x,C$_{PRE}$]

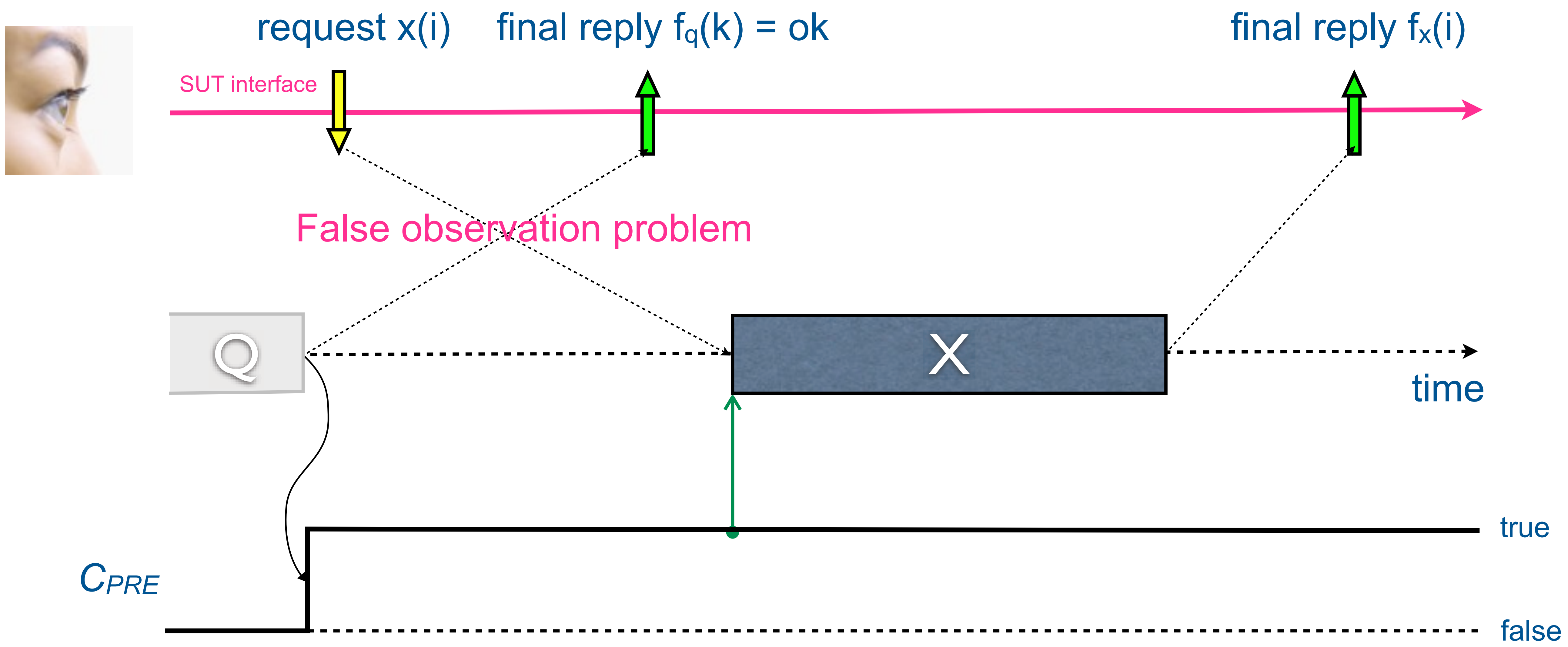
$C_{PRE}(x(i)) = \exists\, f_q(k),\ [t(f_q(k)) < t(x(i))] \wedge [f_q(k) = ok]$



		$f_x(i)$			
		$\in \{Z_x, T_x\}$	r_P	$\in R_x \setminus r_P$	ø
$C_{PRE}(x(i))$	true	**TN**	**FP**	**OP** (other positive)	w
	false	**FN**	**TP**		

Oracle for property PC[x,C$_{PRE}$]

$$C_{PRE}(x(i)) = \exists\, f_q(k), [t(f_q(k)) < t(x(i))] \wedge [f_q(k) = ok]$$



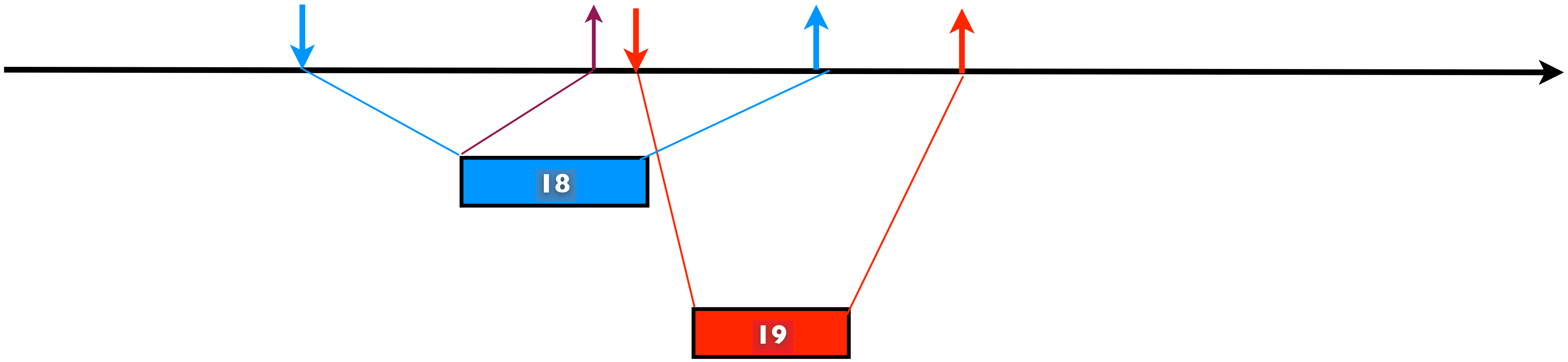
		$f_x(i)$			
		$\in \{Z_x, T_x\}$	r_P	$\in R_x \setminus r_P$	ø
$C_{PRE}(x(i))$	true	**TN**	**FP**	**OP** (other positive)	w
	false	**FN**	**TP**		

Excerpt of trace #137

137 PRE False Neg

```
1290097074.22    send  18    ASPECT_SETDYNAMICSEGSSOURCE
1290097074.27    ir    18              ASPECT_SETDYNAMICSEGSSOURCE
1290097074.31    send  19    ASPECT_ASPECTFROMPOSTERCONFIG
1290097074.37    rcv   18              ASPECT_SETDYNAMICSEGSSOURCE  OK
1290097099.64    rcv   19             ASPECT_ASPECTFROMPOSTERCONFIG      S_aspect_stdGenoM_ACTIVITY_INTERRUPTED
```

19 should apparently be rejected since reply 18 not yet received, but it isn't : false observation



Robustness over traces

	Total traces	FN traces	FP traces	Hung traces	Bad traces	Robust traces
GenoM	293	74	5		76	74,06 %
BIP-A	293	40		42	80	72,7 %
BIP-B	293	11		1	12	95,9 %

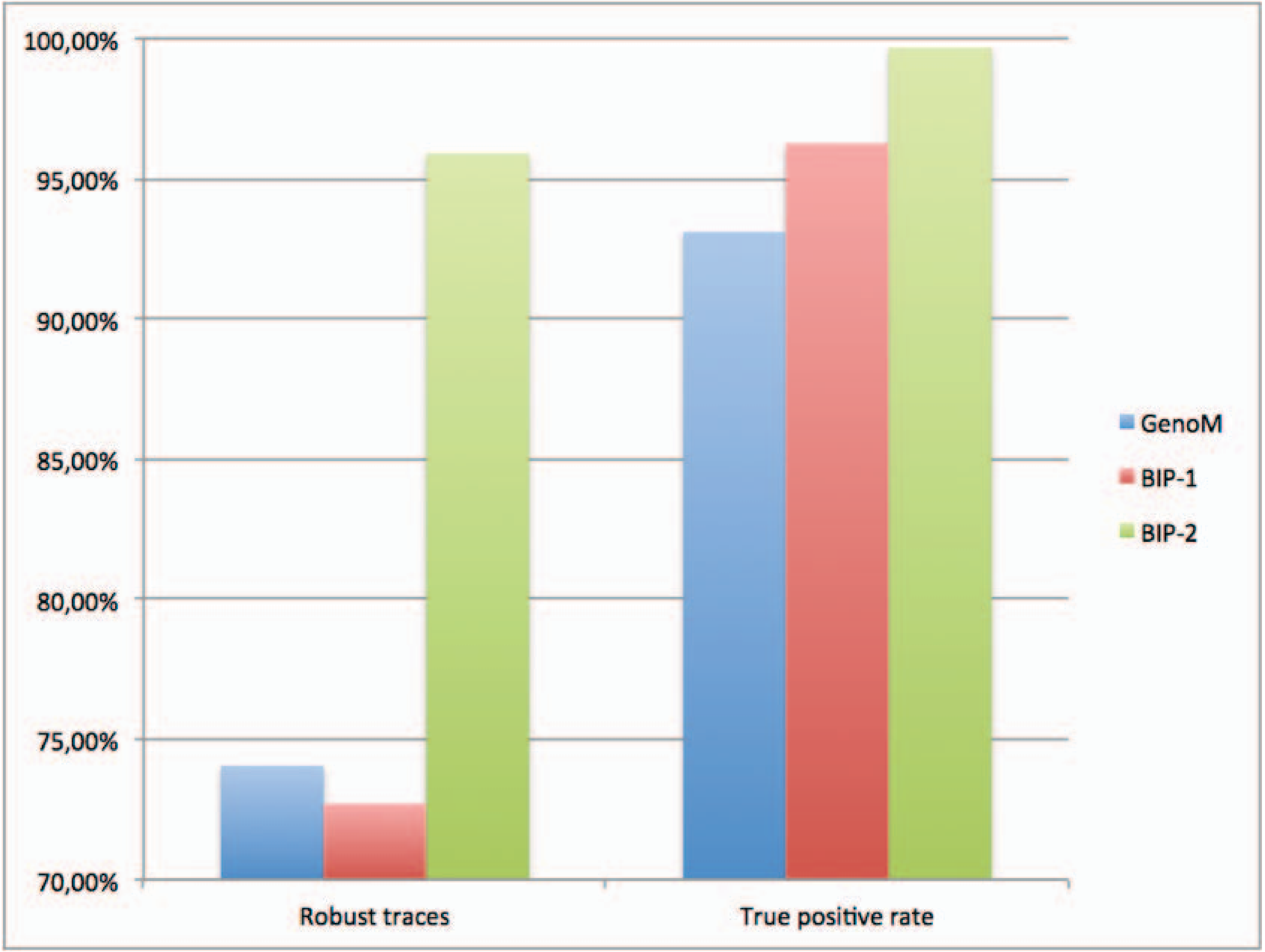
Robustness over properties

True positive rate (coverage) = TP / (TP + FN)

False positive rate = FP / (FP + TN)

	Tests	TN	TP	FN	FP	W	TPR	FPR
GenoM	34780	29142	5112	379	32	115	93,1 %	0,1 %
BIP-A	30955	26036	4495	175		249	96,3 %	0,0 %
BIP-B	35066	29226	5694	19		127	99,7 %	0,0 %

Overall Comparison



Conclusion

- **Testing was a useful complement to formal development using BIP since people, tools and run-time environments are not correct-by-construction**
- **Timing robustness property oracles more difficult to formulate than expected**
- **Implementation as SQL queries on XML-coded traces was a good choice**
- **Black-box timing robustness testing**
  - ☺ Benchmarking of different implementations
  - ☹ Possibility of false observations