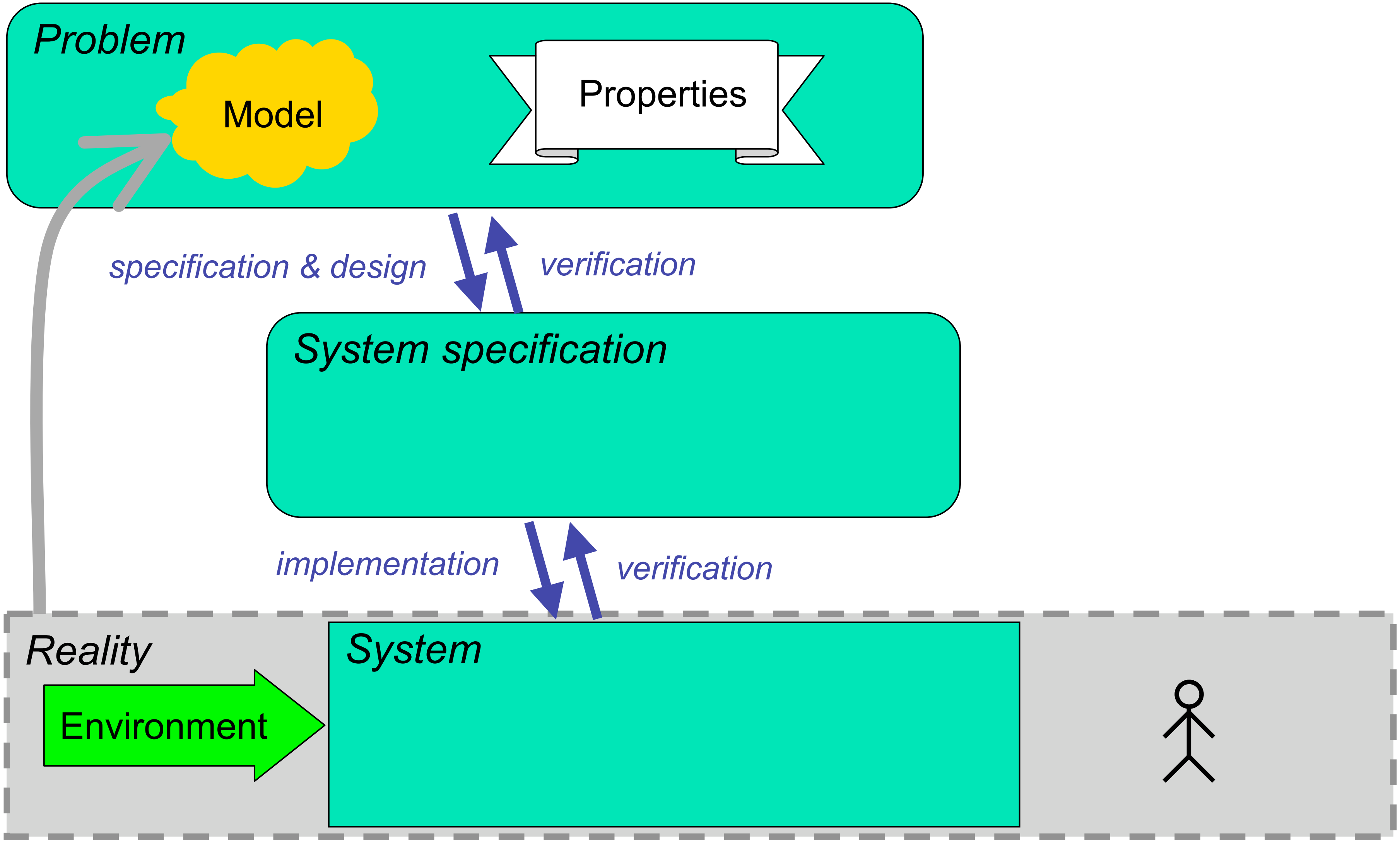
Matching Distributed System Models to Reality

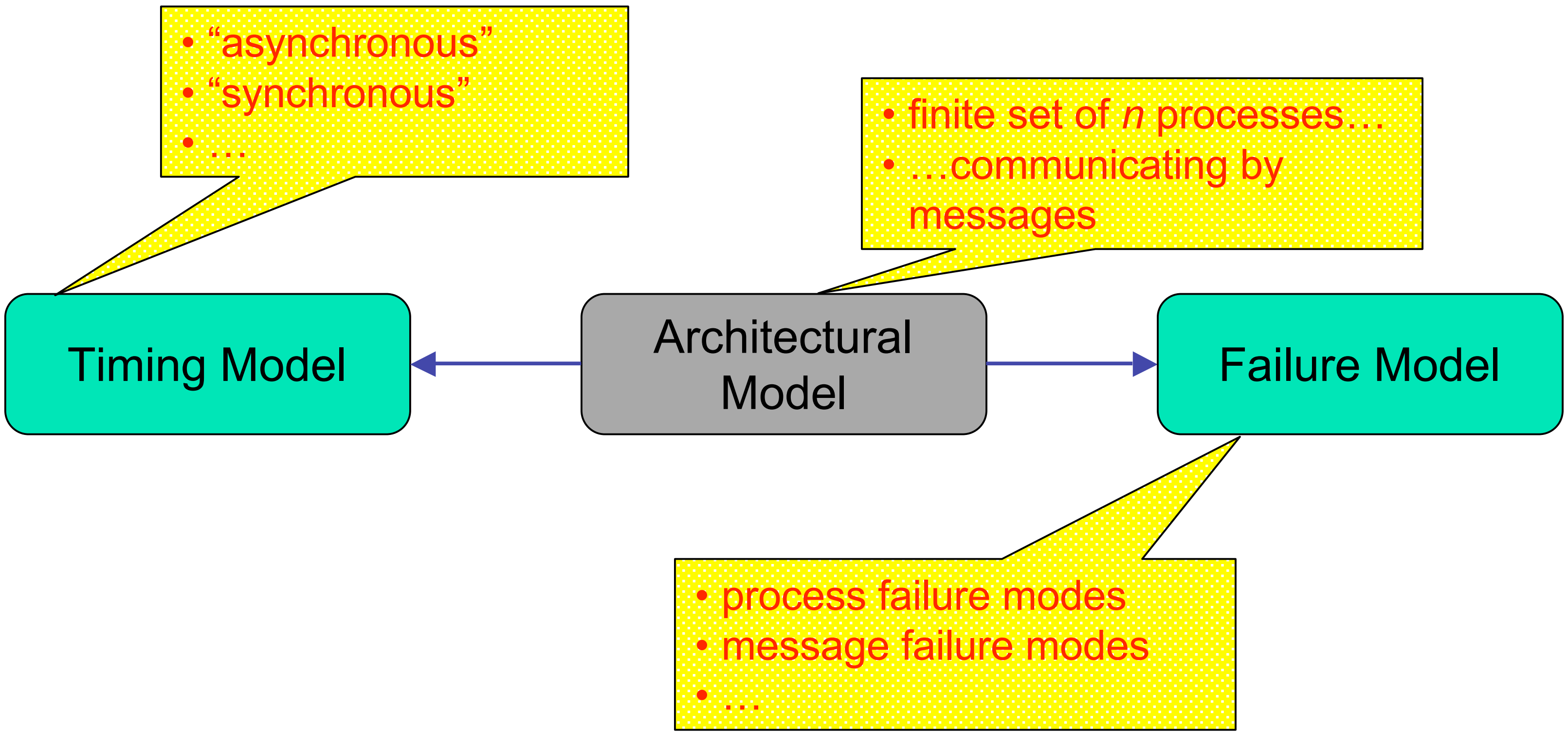
David Powell
LAAS-CNRS

DISC, Amsterdam, 4-7.10.2004

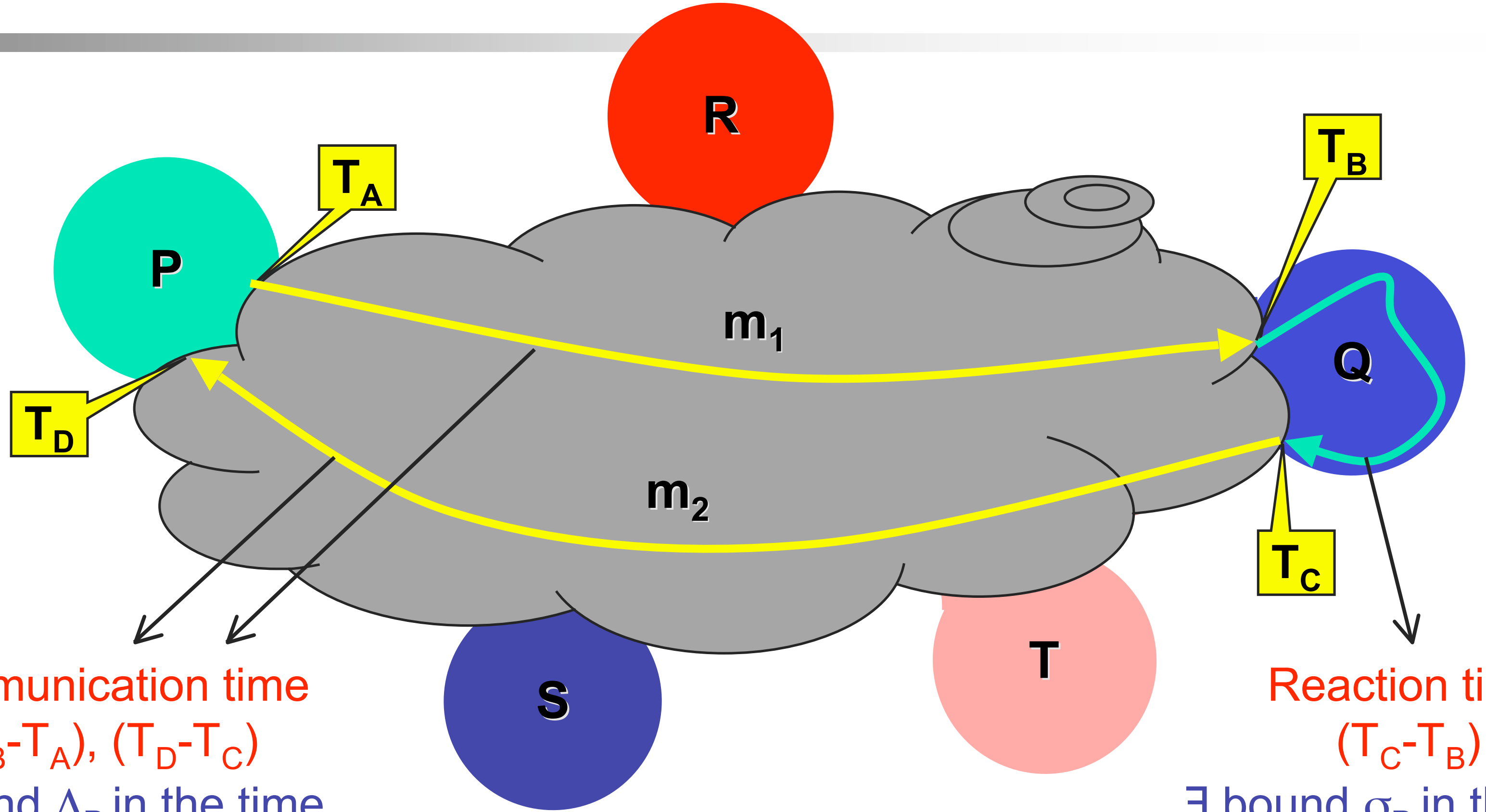
From Abstraction to Reality



Distributed System Models



Timing Models



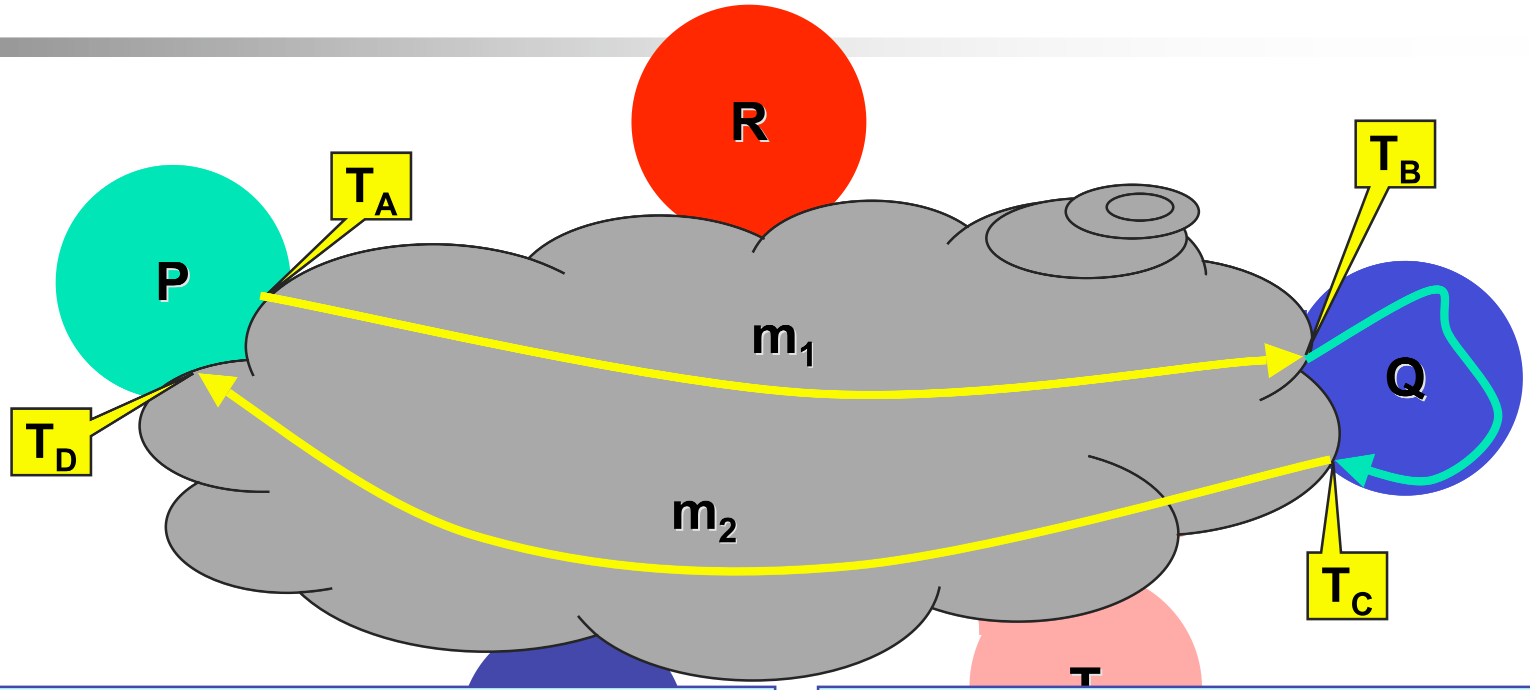
Communication time
(T_B-T_A), (T_D-T_C)
∃ bound Δ_P in the time reference of P?

Reaction time
(T_C-T_B)
∃ bound σ_P in the time reference of P?

Both bounds must be defined so that P can detect that *something* has failed

One bound must be guaranteed so that P can decide *what* has failed

Timing Models



Time-free

- either communication or reaction time bound is not defined
- P cannot decide if Q has stopped, or if Q, m1 or m2 are very slow

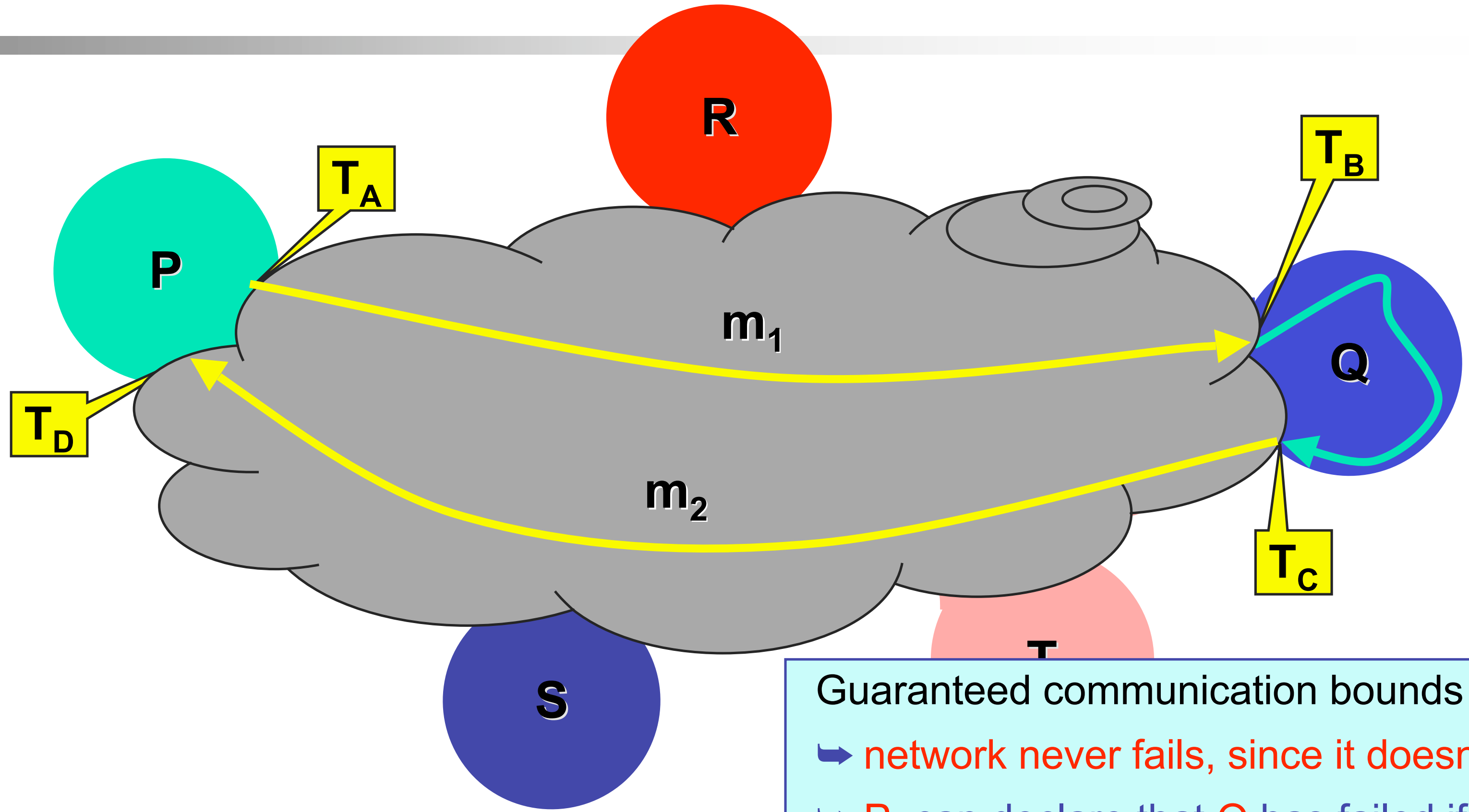
Cannot (deterministically) solve consensus and other agreement problems

Guaranteed communication bounds

- communication bound guaranteed (the network never fails)
- P can declare that Q has failed if $T_D - T_A > 2\Delta_P + \sigma_P$

Confidence?

Timing Models



Irrefutable justification of guaranteed communication bounds:

- *each process has a private network* (a single fault confinement region)

Guaranteed communication bounds

- network never fails, since it doesn't exist!
- P can declare that Q has failed if $T_D - T_A > 2\Delta_P + \sigma_P$

Total confidence

Timing Models

Bounds \ Guarantees	No	Soft	Firm
No (NB model)	unreliable asynchronous	fair lossy asynchronous	reliable asynchronous
Unknown (UB model)	?	?	partially synchronous
Known (KB model)	unreliable synchronous	eventually synchronous	reliable synchronous

[Le Lann *et al.* 1994]

Failure Models

- Time domain
  - none
  - stopping
  - omission
  - timing (KB model only)
    - early
    - late
  - arbitrary (or undefined)
- Value domain
  - none
  - non-code (signaled)
  - arbitrary (non-signaled)
    - ↪ data
    - ↪ meta-data
      - data sender
      - data originator
      - data creation time
      - …

process crash model

Failure Models

- Time domain
  - none
  - stopping
  - omission
  - timing (KB model only)
    - early
    - late
  - arbitrary (or undefined)
- Value domain
  - none
  - non-code (signaled)
  - arbitrary (non-signaled)
    - ↪ data
    - ↪ meta-data
      - data sender
      - data originator
      - data creation time
      - …

arbitrary failure model

Failure Models

- Time domain
  - none
  - stopping
  - omission
  - timing (KB model only)
    - early
    - late
  - arbitrary (or undefined)
- Value domain
  - none
  - non-code (signaled)
  - arbitrary (non-signaled)
    - ↪ data
    - ↪ meta-data
      - data sender
      - data originator
      - data creation time
      - …

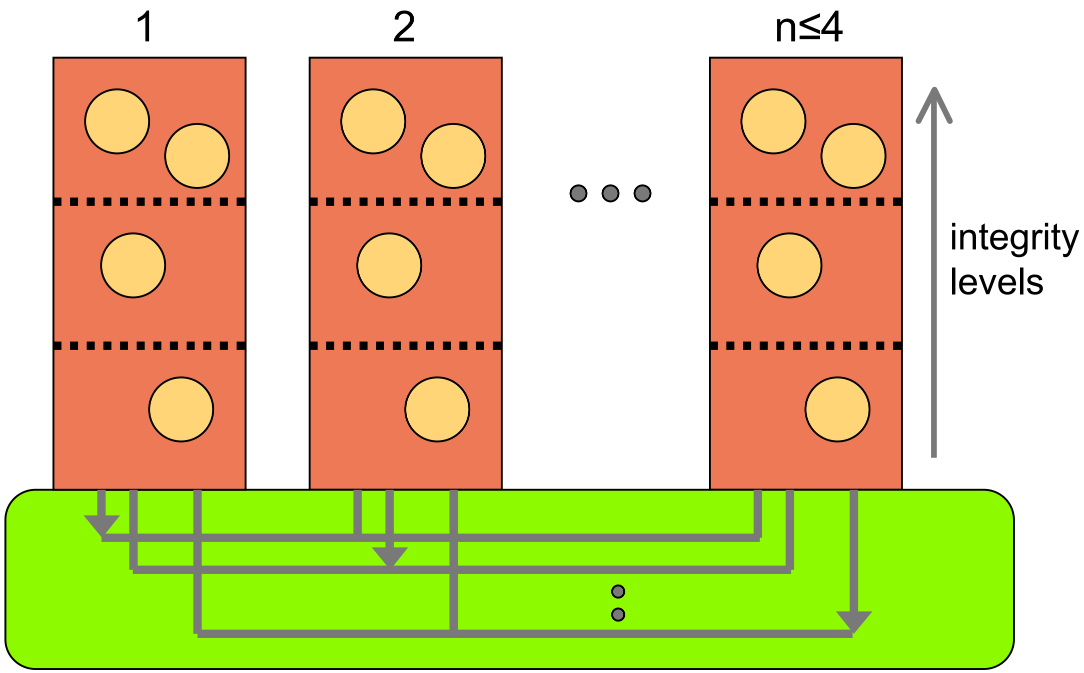
authenticated arbitrary failure model

Example Systems

- GUARDS (1996-1999)
  - embedded system for space, railways, nuclear propulsion
  - permanent & transient physical faults, design faults
- Delta-4 (1986-1991)
  - factory automation, business systems
  - permanent & transient physical faults, intrusions
- MAFTIA (2000-2002)
  - Internet security
  - intrusions, permanent physical faults
- PADRE (1994-1997)
  - railway automation
  - permanent & transient physical faults

GUARDS

[Powell *et al.* 1999]



- embedded system for space, railways, nuclear propulsion
- permanent & transient physical faults, design faults

Process failure model

n=4 Arbitrary ①
n=3 Arbitrary + authentication ①
↪ *keyed CRC*
n=2 Crash ②
↪ *self-checking*

Timing model

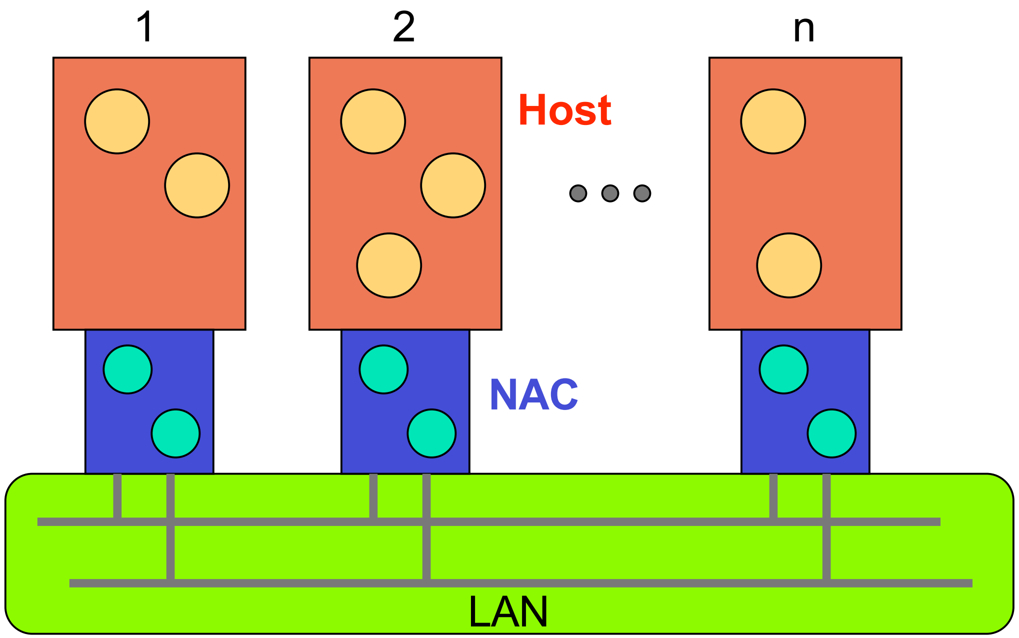
Reliable synchronous
↪ *private channels*

FT Services

- Clock synchronization
- Interactive consistency
- Active replication
  ① with or
  ② without voting
- …

Delta-4

[Powell 1994]



- factory automation, business systems
- permanent & transient physical faults, intrusions

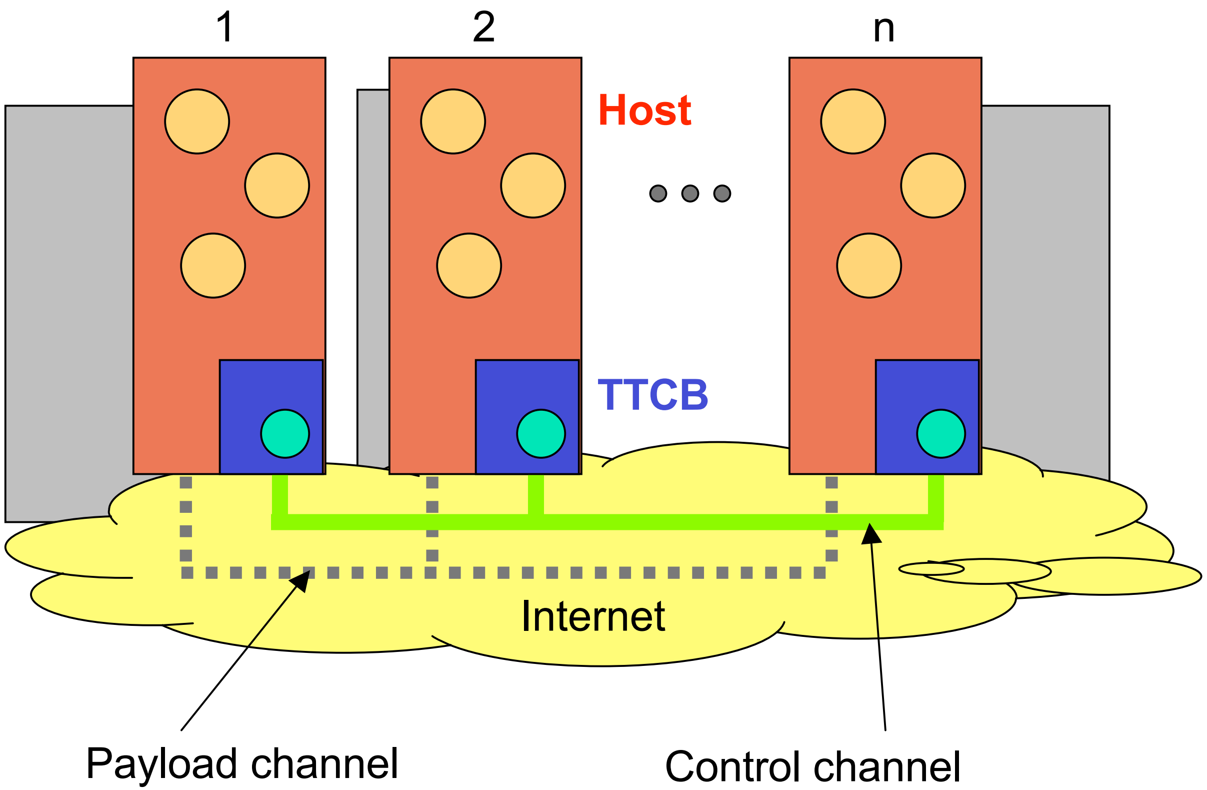
Process failure model (hybrid)
Hosts: ① Arbitrary
② Crash
↪ *self-checking*
NACs: Crash
↪ *self-checking*

Timing model
Reliable synchronous
↪ *bounded omission faults*
↪ *bounded channel faults*

FT Services
- Atomic multicast
- Active replication
  ① with or
  ② without voting
- ② Passive replication
- ② Semi-active replication
- …

MAFTIA

[Verissimo *et al.* 2004]



- Internet security
- intrusions, permanent physical faults

Process failure model

Hosts: Arbitrary + authent. ①
↪ *threshold crypto.*
TTCB: Crash ②
↪ *self-checking*
↪ *tamperproof*

Timing model

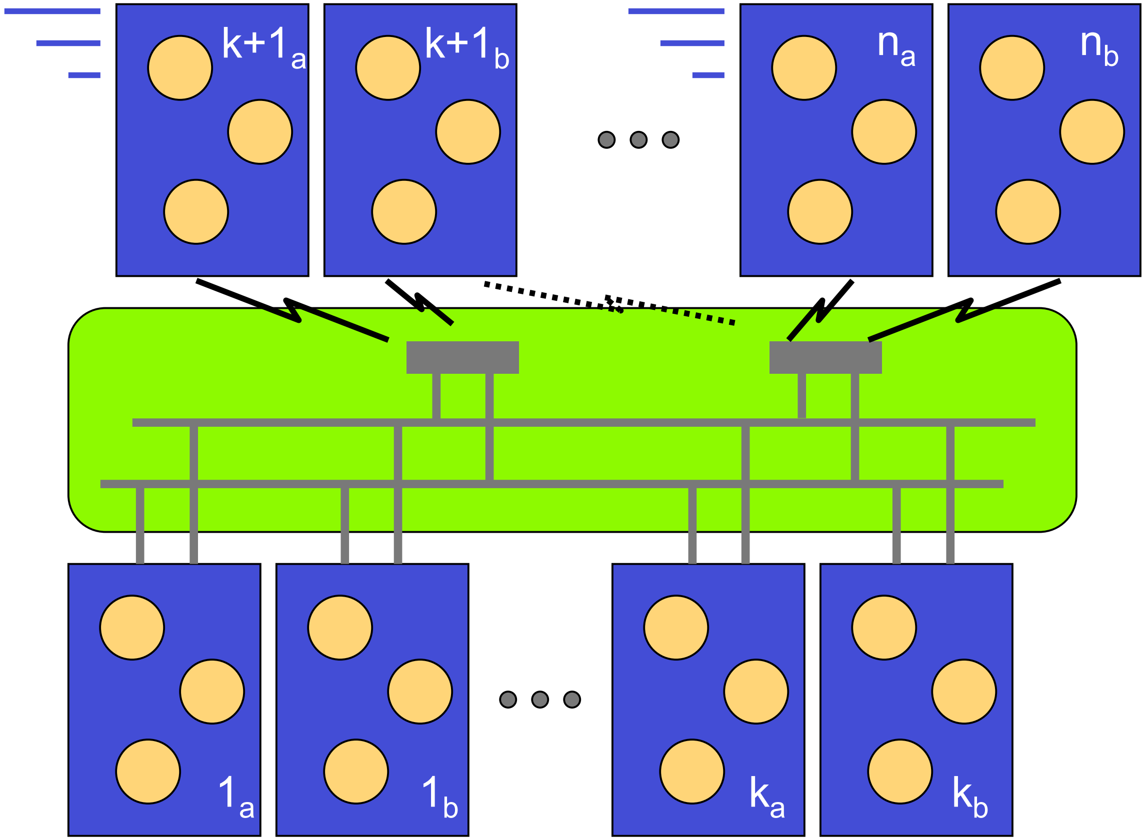
Hosts / Payload:
Reliable asynchronous ①
TTCB / Control:
Reliable synchronous ②
↪ *tamperproof reserved chan.*

FT Services

① Randomized binary agreement
① Atomic broadcast
① + ② Block agreement
① + ② Reliable multicast
…

PADRE

[Essamé *et al.* 1999]



Process failure model

Crash

↪ *self-checking (coded processor technique)*

Timing models

Safety

- Base - unreliable synchronous
- Derived - ‘safe synchronous’ (fail-aware datagram)
  ↪ *fail-safe local clocks*

Availability

- Eventually synchronous

FT Service

Fail-safe duplex redundancy

↪ *fail-safe exclusion relay*

- railway automation
- permanent & transient physical faults

Assumption Coverage

[Powell 1992]

- Measure of confidence in an assumption
- Likelihood that assumption holds true in given universe (sample set)
- Sets upper bound on dependability

$$\Pr\left\{\begin{array}{c|c} \textit{system} & \textit{real} \\ \textit{property} & \textit{system} \end{array}\right\} = \Pr\left\{\begin{array}{c|c} \textit{system} & X \\ \textit{property} & \end{array}\right\} \times \Pr\left\{\begin{array}{c|c} X & \textit{real} \\ & \textit{system} \end{array}\right\} + \varepsilon$$

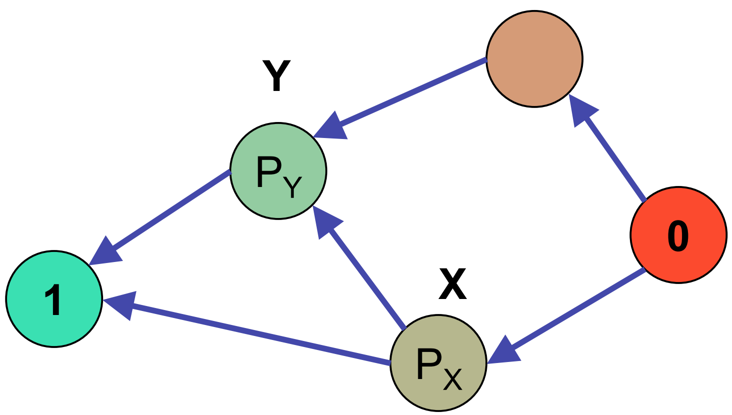
likelihood that system property holds under assumption(s) *X*

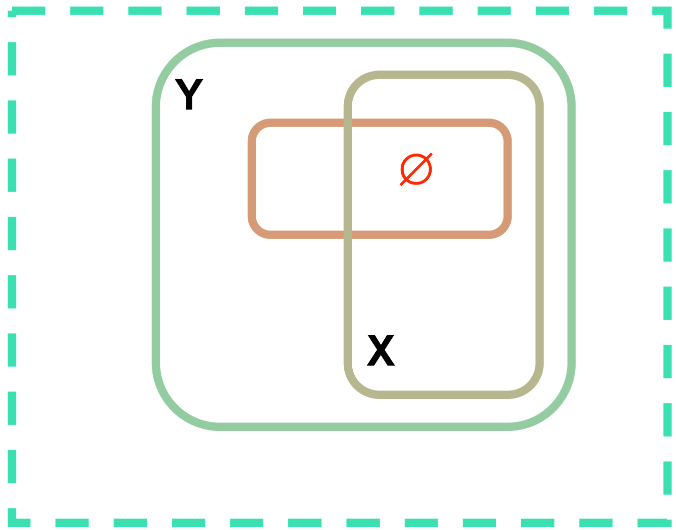
coverage of assumption(s) *X* $\hookrightarrow P_X$

Assumption Ranking

[Powell 1992]

- ↗ General = ↗ Permissive = ↗ Coverage
- If $X \Rightarrow Y$ (equivalently $Y \supseteq X$), then $P_Y \geq P_X$

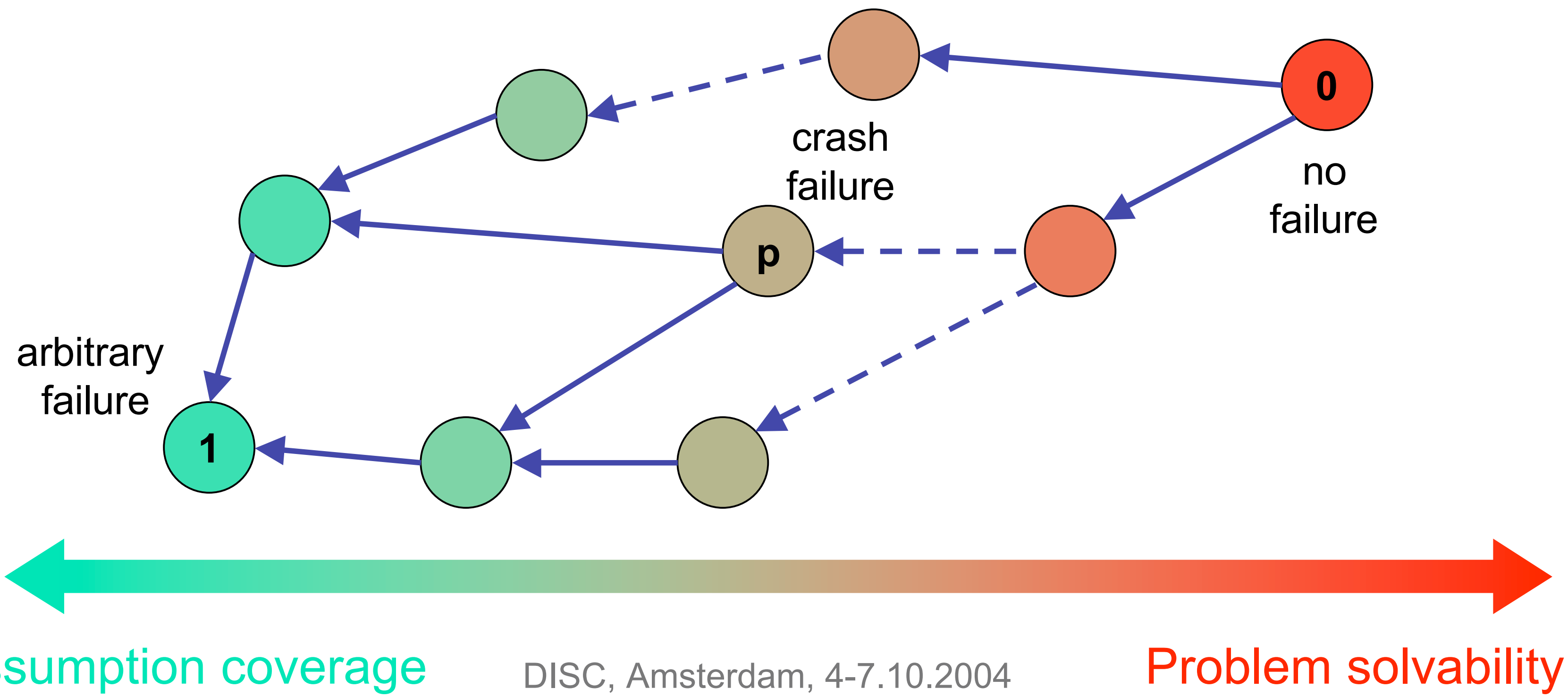




Assumption Ranking

[Powell 1992]

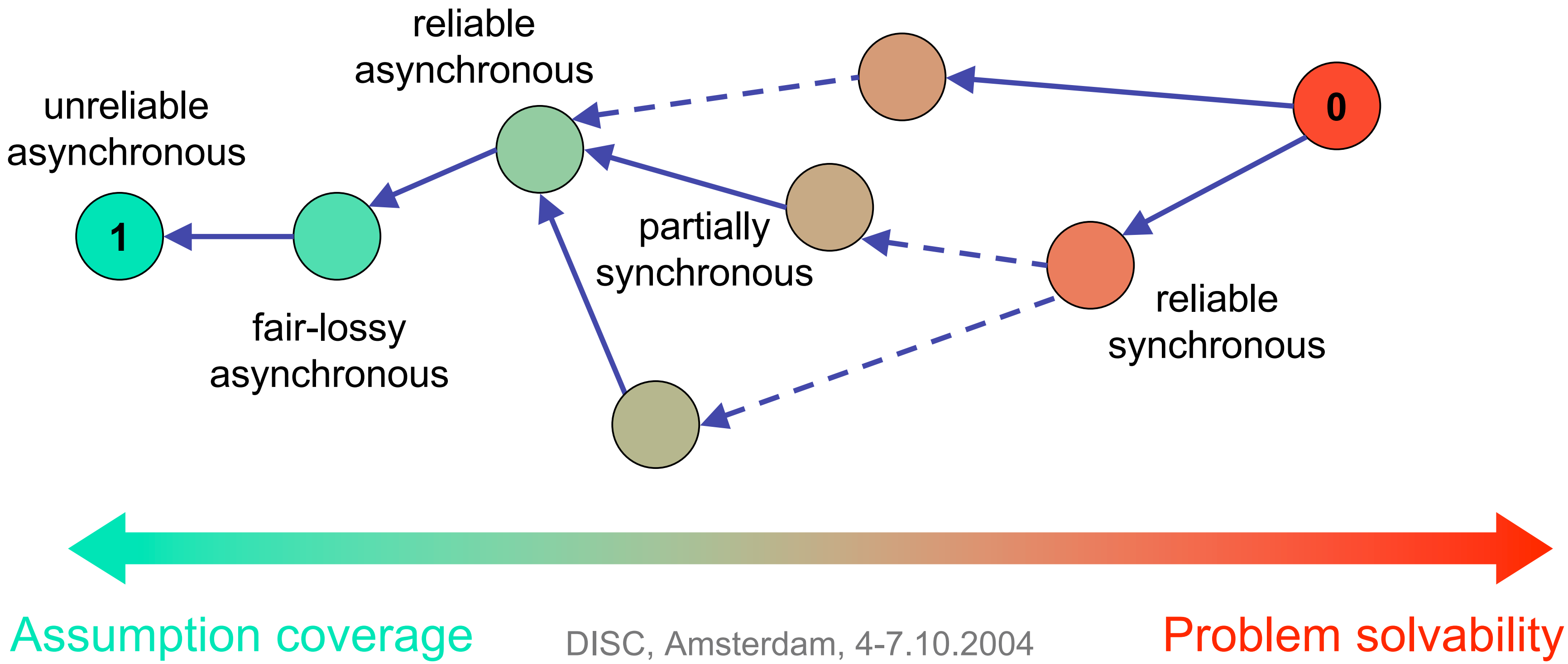
- ↗ General = ↗ Permissive = ↗ Coverage
- If $X \Rightarrow Y$ (equivalently $Y \supseteq X$), then $P_Y \geq P_X$



Assumption Ranking

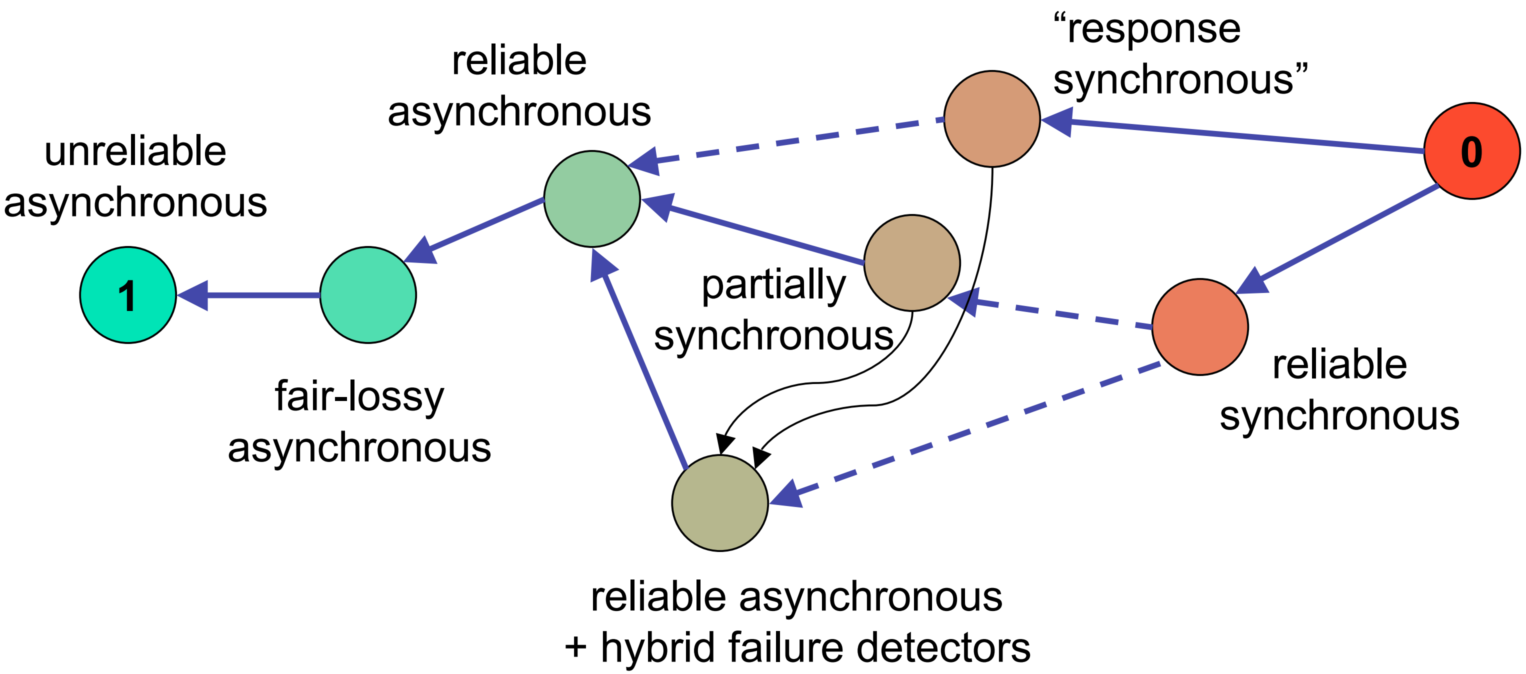
[Powell 1992]

- ➚ General = ➚ Permissive = ➚ Coverage
- If $X \Rightarrow Y$ (equivalently $Y \supseteq X$), then $P_Y \geq P_X$



Alternative Assumptions

- If $X = A \cup B$ then $P_x = P_A + P_B - P_{A \cap B}$
- Alternate base models $\Rightarrow P_x \geq \max(P_A ; P_B)$



Linking to Dependability Assessment

Define $E^t \equiv \{E(\tau), \tau \in [0,t]\}$ and $R_E(t) = \Pr\{E^t\}$

With C the (composite) system property defining "correct"

then $R_C(t)$ is a measure of system reliability

If $X = \bigcap_i H_i$ denotes the system model assumed to prove C

we can write: $R_C(t) \le R_X(t)$ → "assumption reliability" [Latronico *et al.* 2004]

Example:

- H_0 — finite set of n processes
- H_1 — processes fail only by crashing
- H_2 — at most k processes fail
- H_3 — all message delays < Δ

Towards Dependability Assessment

- H_0 — finite set of n processes
- H_1 — processes fail by crashing
- H_2 — at most k processes fail
- H_3 — all message delays < Δ

$$R_X(t) = \Pr\left\{H_0^t \cap H_1^t \cap H_2^t \cap H_3^t\right\}$$

$$= \Pr\left\{H_0^t \cap H_1^t \cap H_2^t\right\} \cdot \Pr\left\{H_3^t\right\}$$

(assuming stochastic independence of H_3^t)

$$= \Pr\left\{H_0^t\right\} \cdot \Pr\left\{H_1^t \cap H_2^t \middle| H_0^t\right\} \cdot \Pr\left\{H_3^t\right\}$$

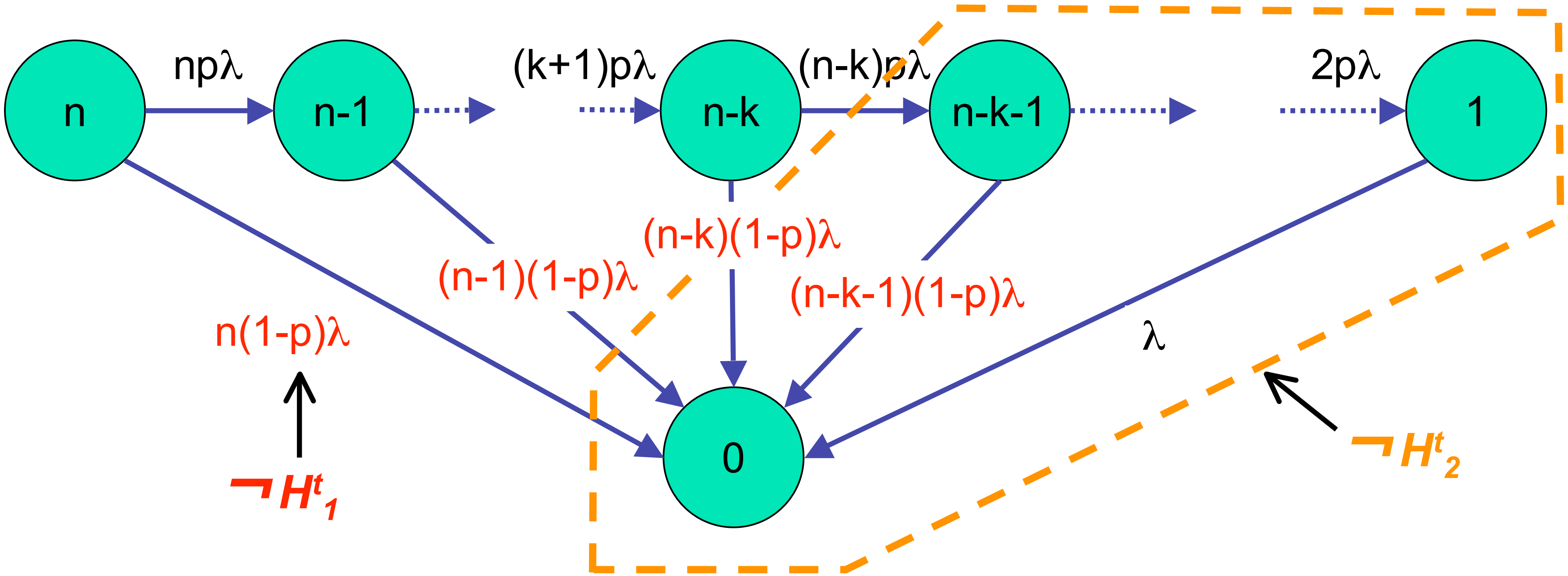
- $\Pr\{H_0^t\}$ → =1 (axiom)
- $\Pr\{H_1^t \cap H_2^t | H_0^t\}$ → system state transition model
- $\Pr\{H_3^t\}$ → communication model, e.g. $\left[(1-q)F(\Delta)\right]^{M(t)}$

Impact of Assumption Coverage

Consider *n*-unit system tolerating *k* faults

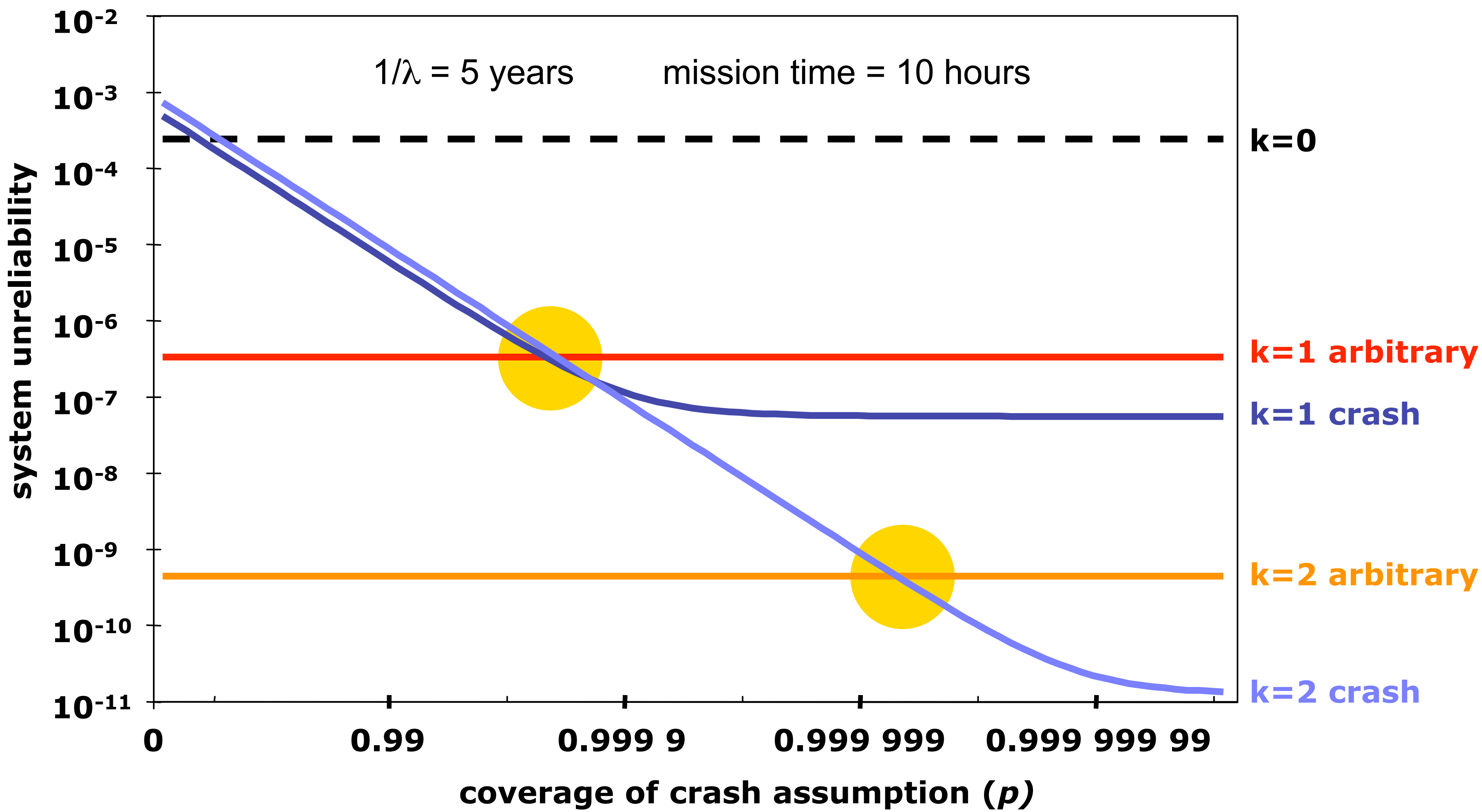
- ***H_1*** — processes fail by crashing
- ***H_2*** — at most *k* processes fail

	Crash	Arbitrary
	$p<1$ $n \geq k+1$	$p=1$ $n \geq 3k+1$
$k=0$	$n=1$	$n=1$
$k=1$	$n=2$	$n=4$
$k=2$	$n=3$	$n=7$



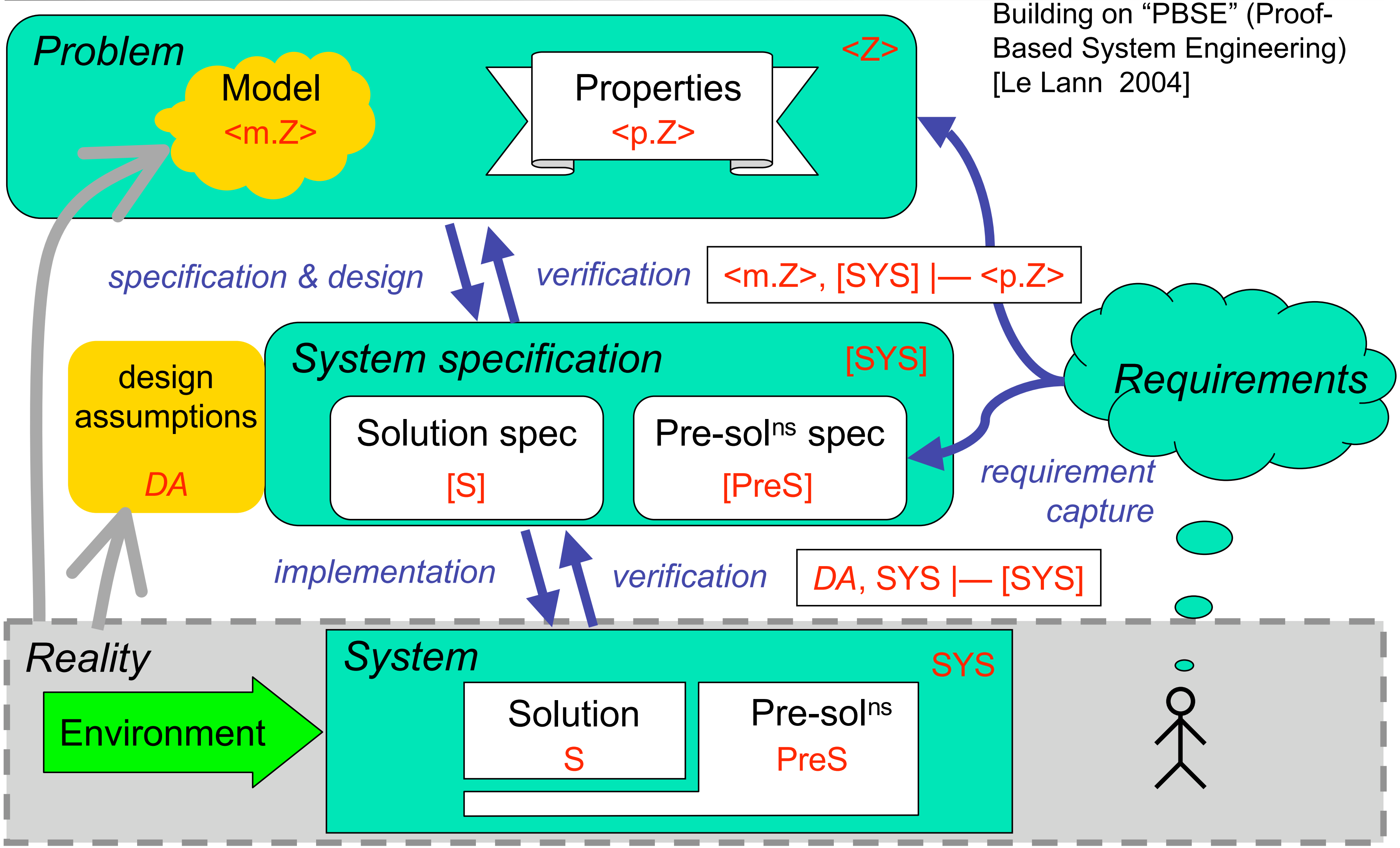
Impact of Assumption Coverage

[Powell 1992]



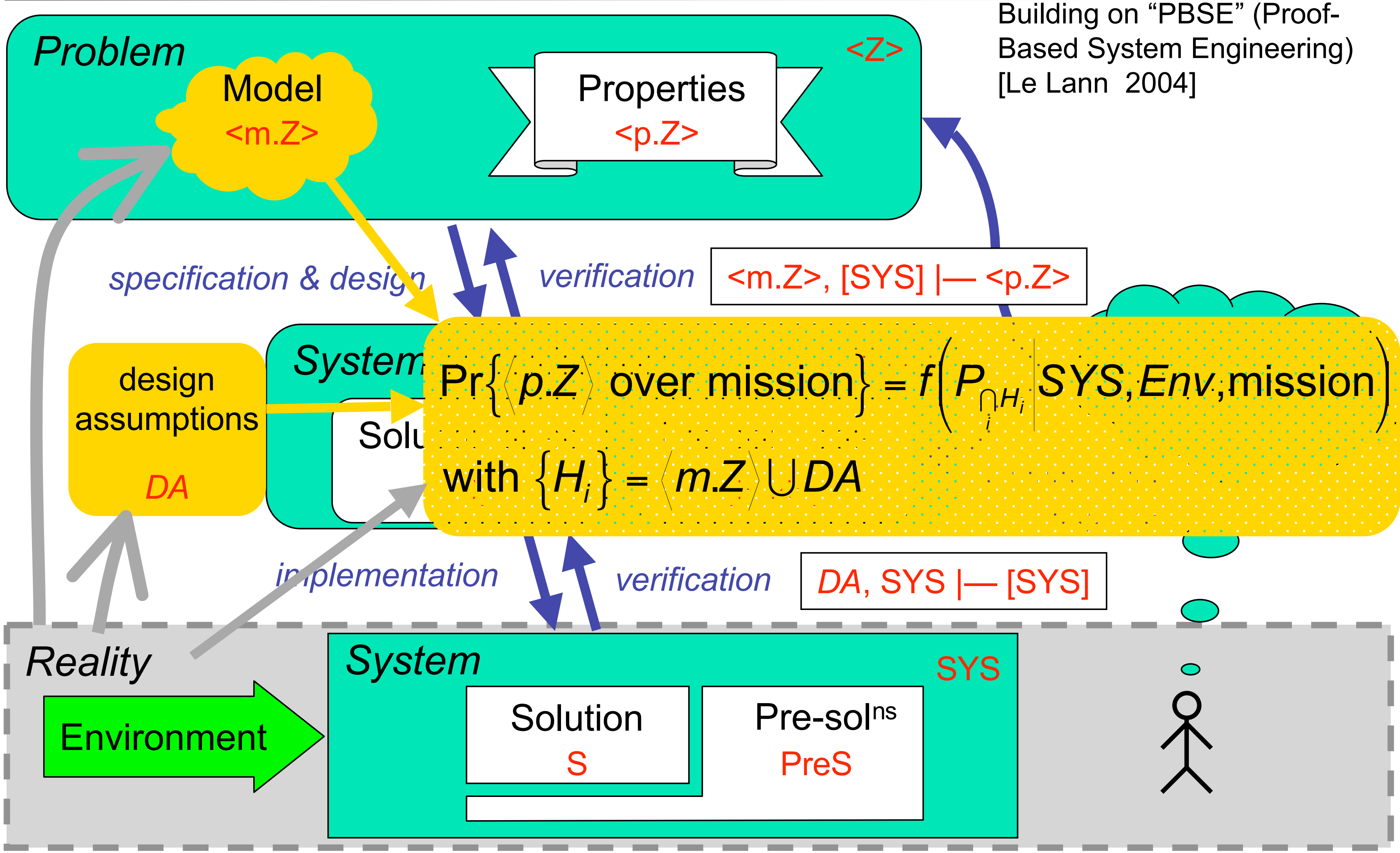
Coverage in System Engineering

Building on “PBSE” (Proof-Based System Engineering) [Le Lann 2004]



Coverage in System Engineering

Building on “PBSE” (Proof-Based System Engineering) [Le Lann 2004]



Conclusions (1/3)

- Valid model has compatible sub-models
- Good model has permissive sub-models
- Best model depends on:
  - real system in real environment
  - required application-level properties
- Validity of model vs. reality
  - depends on validity of root assumptions
  - captured by assumption coverage

Conclusions (2/3)

- Assumption coverage $\Rightarrow$ upper bounds on stochastic measures of dependability
  - ranges of parameters allowing objectives to be met by given problem/solution pair
  - optimum solution for given problem and range of parameters
- Permissive models
  - higher assumption coverage
  - not necessarily higher dependability

Conclusions (3/3)

- Need:
  - explicit & clear statements of root assumptions
  - method for linking design to assessment through coverage of root assumptions
  - extended distributed system models suitable for current and future real systems (mobility…)

References

- [Essamé et al. 1999] D. Essamé, J. Arlat and D. Powell, "PADRE: a Protocol for Asymmetric Duplex REdundancy", in *Dependable Computing for Critical Applications (DCCA-7)*, (San Jose, CA, USA), January 1999).
- [Latronico et al. 2004] E. Latronico, P. Miner and P. Koopman, "Quantifying the Reliability of Proven SPIDER Group Membership Service Guarantees", in *Dependable Systems and Networks (DSN 2004)*, (Florence, Italy), pp.275-84, 2004.
- [Le Lann et al. 1994] G. Le Lann, P. Minet and D. Powell, "Distributed Systems", in *Fault-Tolerant Computing*, Arago, 15, pp.55-71, O.F.T.A. - Masson, Paris, France, 1994 (in French).
- [Le Lann 2004] G. Le Lann, *Proof-Based System Engineering for Computer-Based Systems: A Guide for Requirement Capture*, version 4.2, INRIA Report, 30 August 2004.
- [Mostefaoui et al. 2004] A. Mostefaoui, D. Powell and M. Raynal, "A Hybrid Approach for Building Eventually Accurate Failure Detectors", in *Pacific Rim Dependable Computing Conference (PRDC'04)*, (Tahiti, French Polynesia), pp.57-65, 2004.
- [Powell 1992] D. Powell, "Failure Mode Assumptions and Assumption Coverage", in *Fault-Tolerant Computing (FTCS-22)*, (Boston, MA, USA), pp.386-95, 1992.
- [Powell 1994] D. Powell, "Distributed Fault-Tolerance — Lessons from Delta-4", *IEEE Micro*, 14 (1), pp.36-47, February 1994.
- [Powell et al. 1999] D. Powell, J. Arlat, L. Beus-Dukic, A. Bondavalli, P. Coppola, A. Fantechi, E. Jenn, C. Rabéjac and A. Wellings, "GUARDS: A Generic Upgradable Architecture for Real-time Dependable Systems", *IEEE Transactions on Parallel and Distributed Systems*, 10 (6), pp.580-99, June 1999.
- [Verissimo et al. 2004] P. Verissimo, N. Neves, C. Cachin, J. Poritz, D. Powell, Y. Deswarte, R. Stroud and I. Welch, Intrusion-Tolerant Middleware: the MAFTIA Approach, LAAS-CNRS, Report 04416, July 2004.