UNIVERSITY OF
Southampton
School of Electronics
and Computer Science

Elements of Foundations for Ubiquitous Computing

the beautiful, the useful and the rest

Vladimiro Sassone

University of Southampton

Ubiquitous Computing: what's that?

Ubiquitous Computing:

computation over a global network of mobile, bounded resources shared among mobile entities which move between highly dynamic, largely unknown, untrusted networks.

Difficulties:

Extreme dynamic reconfigurability; lack of coordination and trust; limited capabilities; partial knowledge . . .

Issues:

Protection and management of resources; privacy and confidentiality of data; . . .

Ubiquitous Computing: what's that?

Ubiquitous Computing:

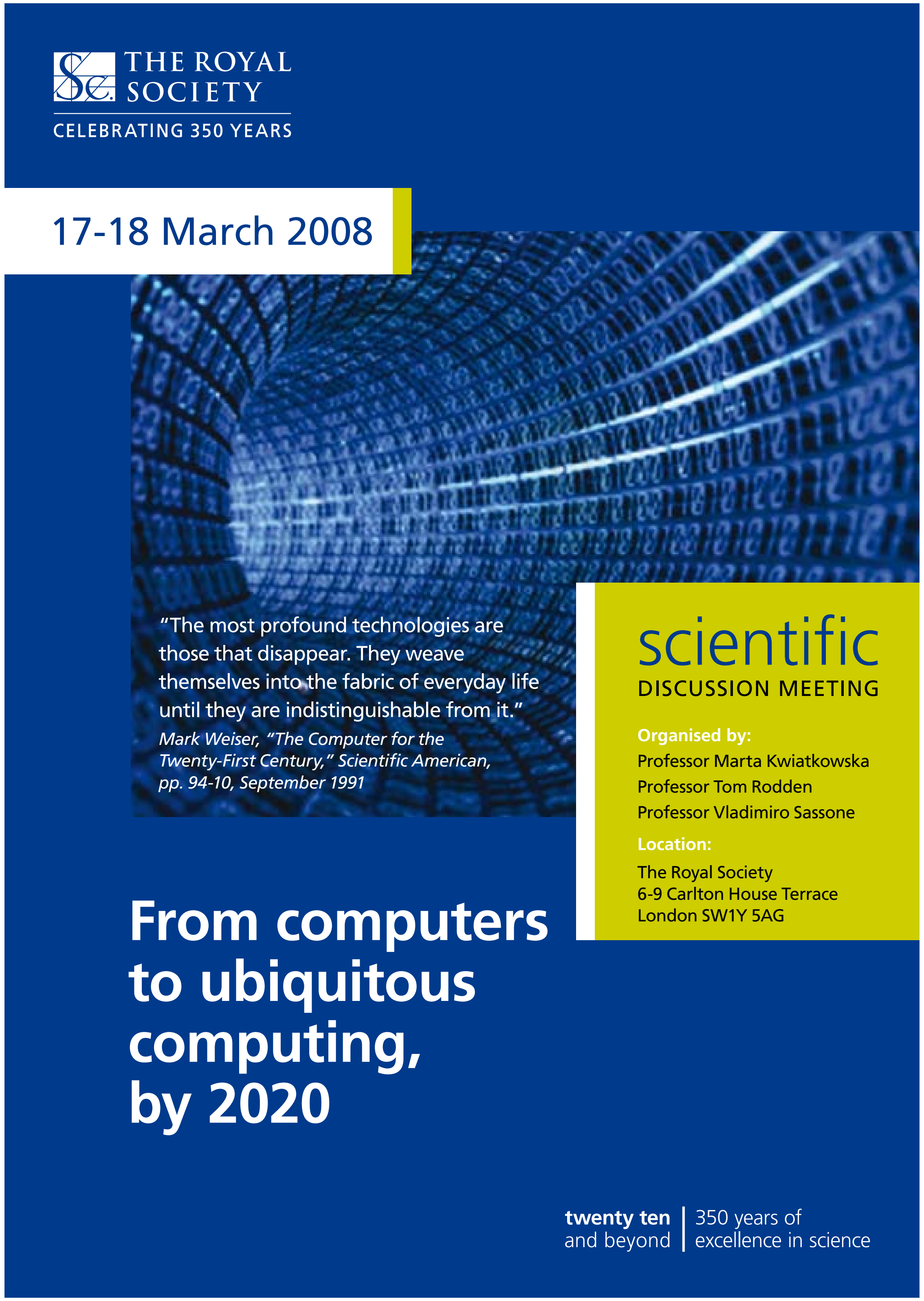
computation over a global network of mobile, bounded resources shared among mobile entities which move between highly dynamic, largely unknown, untrusted networks.

Difficulties:

Extreme dynamic reconfigurability; lack of coordination and trust; limited capabilities; partial knowledge . . .

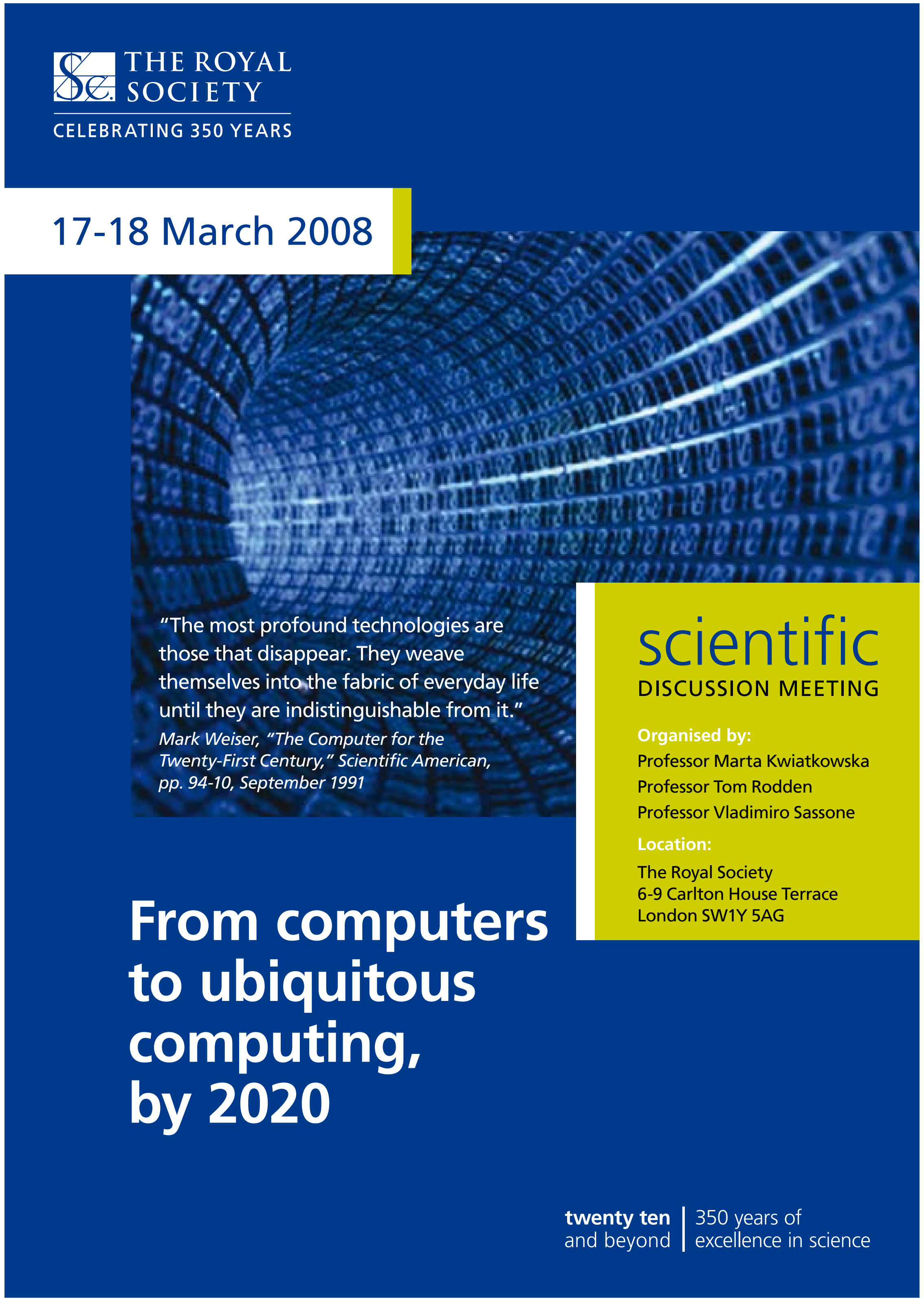
Issues:

Protection and management of resources; privacy and confidentiality of data; . . .



Ubiquitous Computing: what's that?

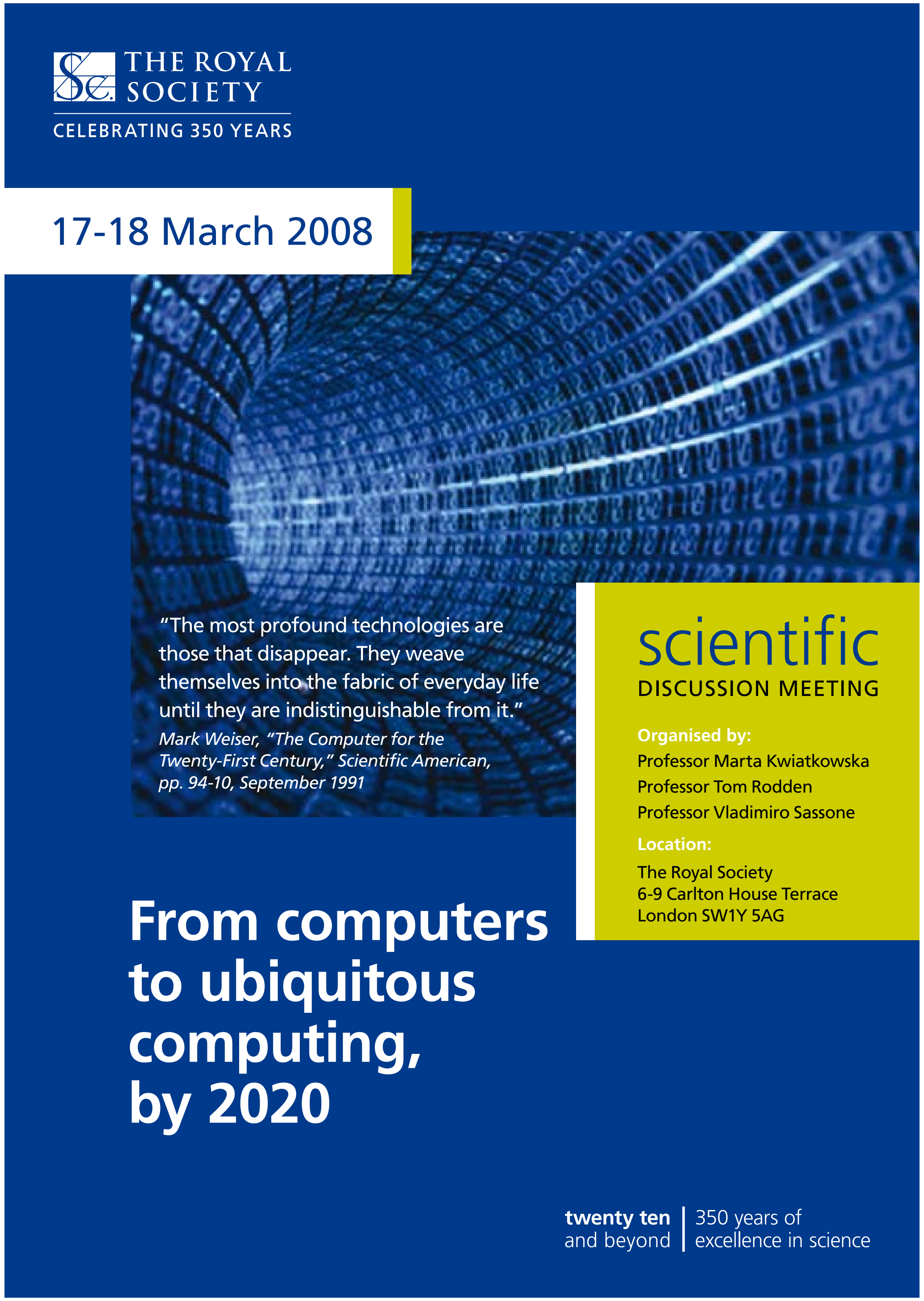
invisible computing



Ubiquitous Computing: what’s that?

invisible computing

sentient computing

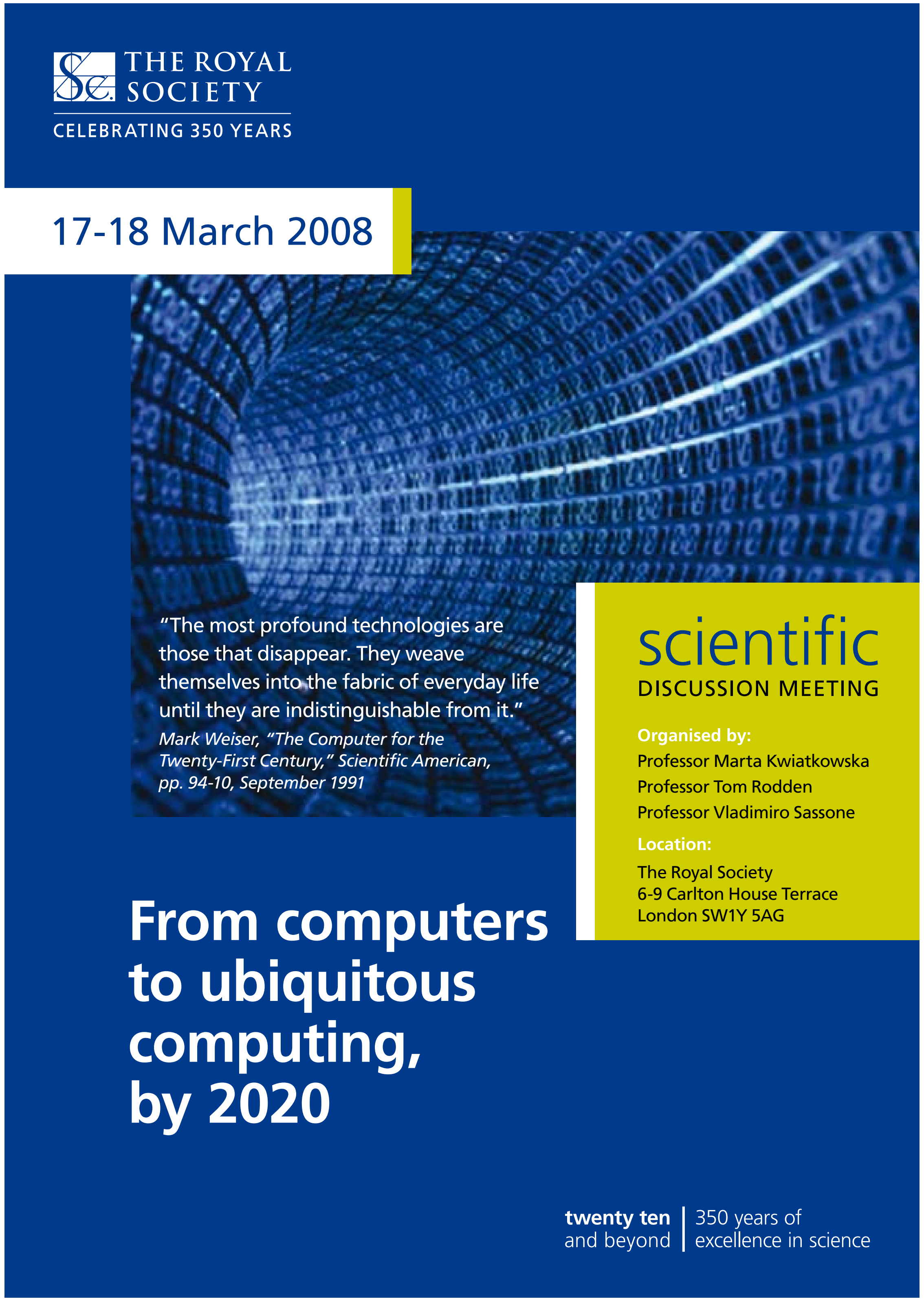


Ubiquitous Computing: what's that?

invisible computing

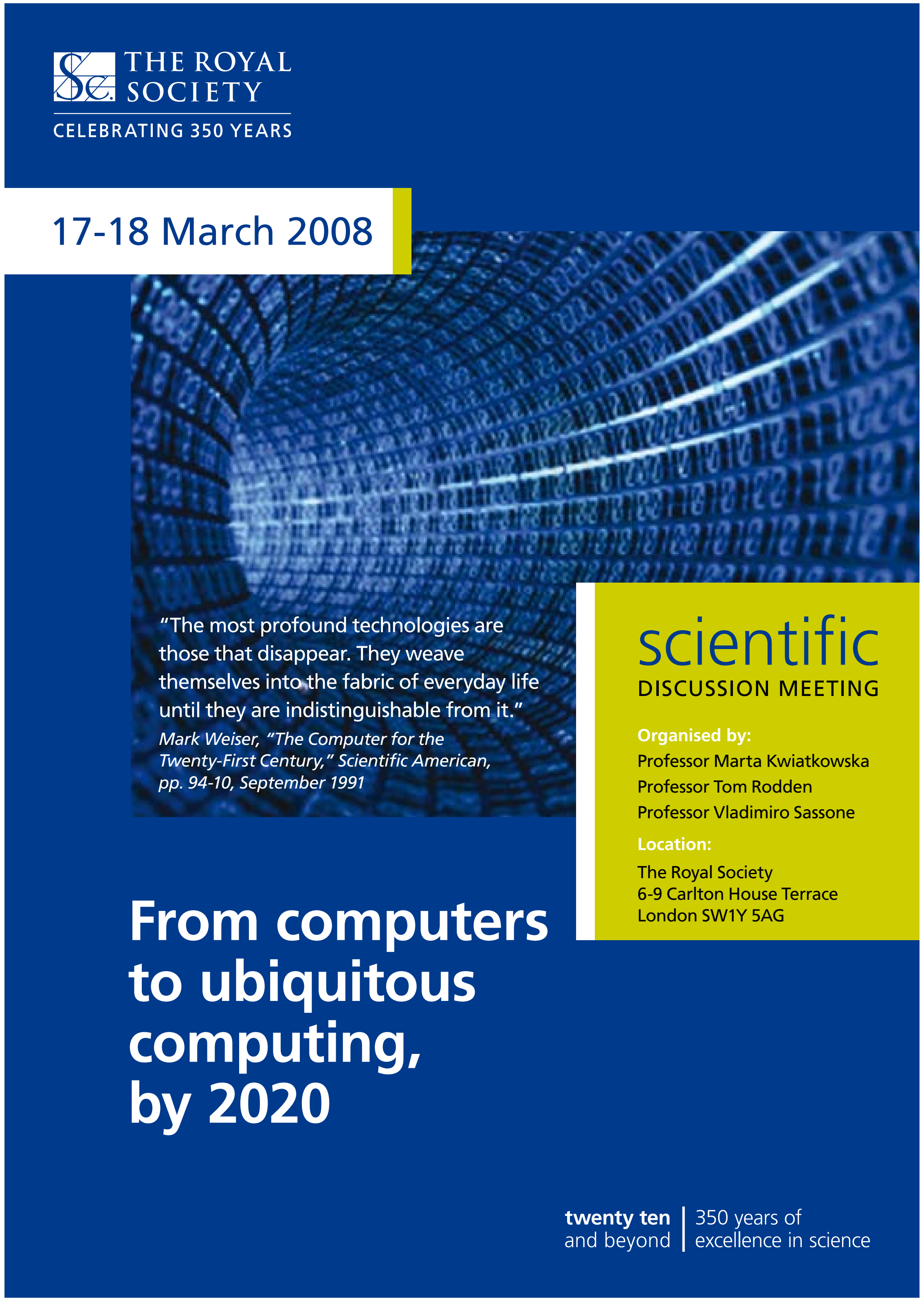
sentient computing

mobile computing



Ubiquitous Computing: what's that?

- invisible computing
- sentient computing
- mobile computing
- autonomic computing



Ubiquitous Computing: what's that?

invisible computing

networks architectures

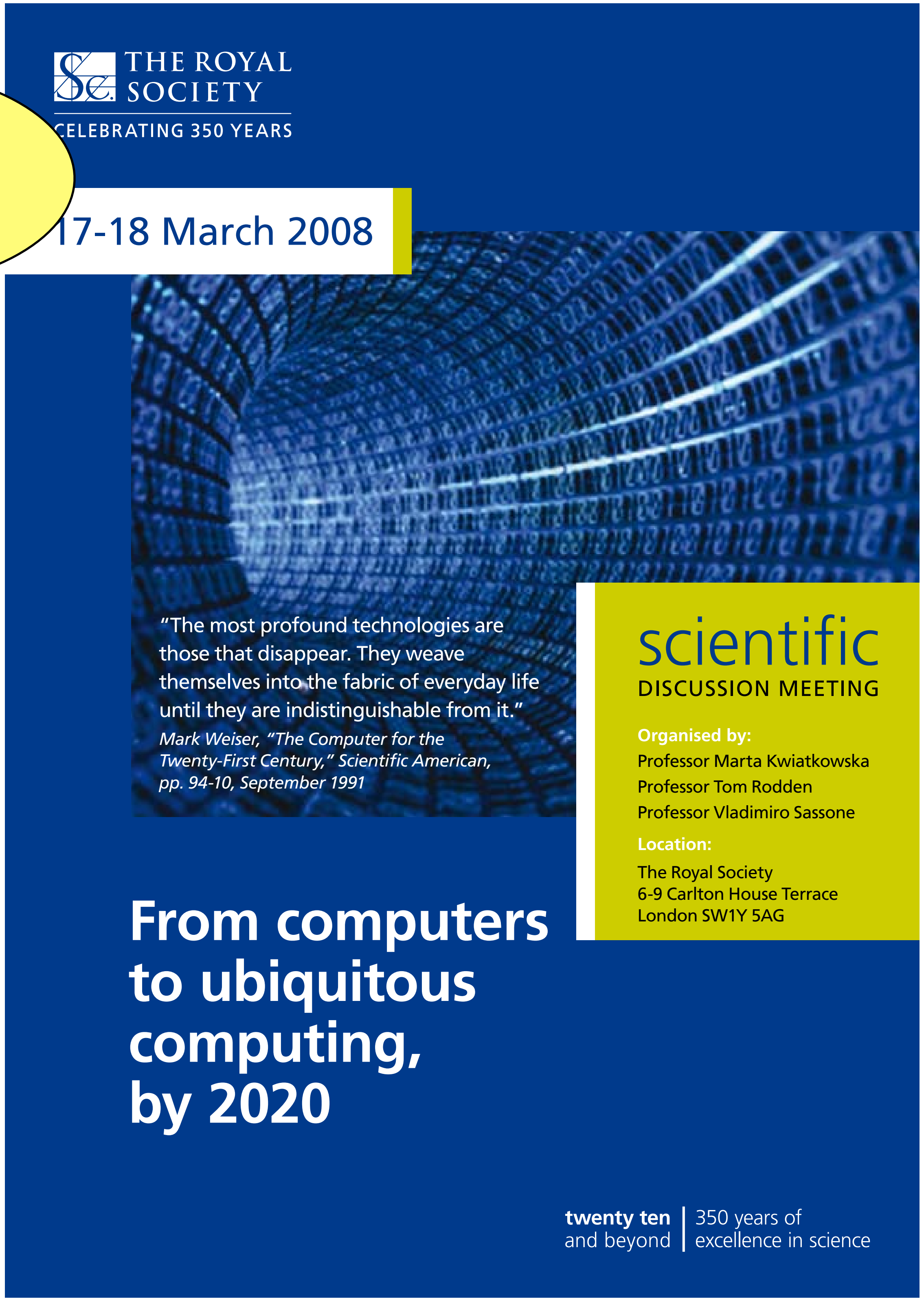
sensor networks

sentient computing

embedded systems

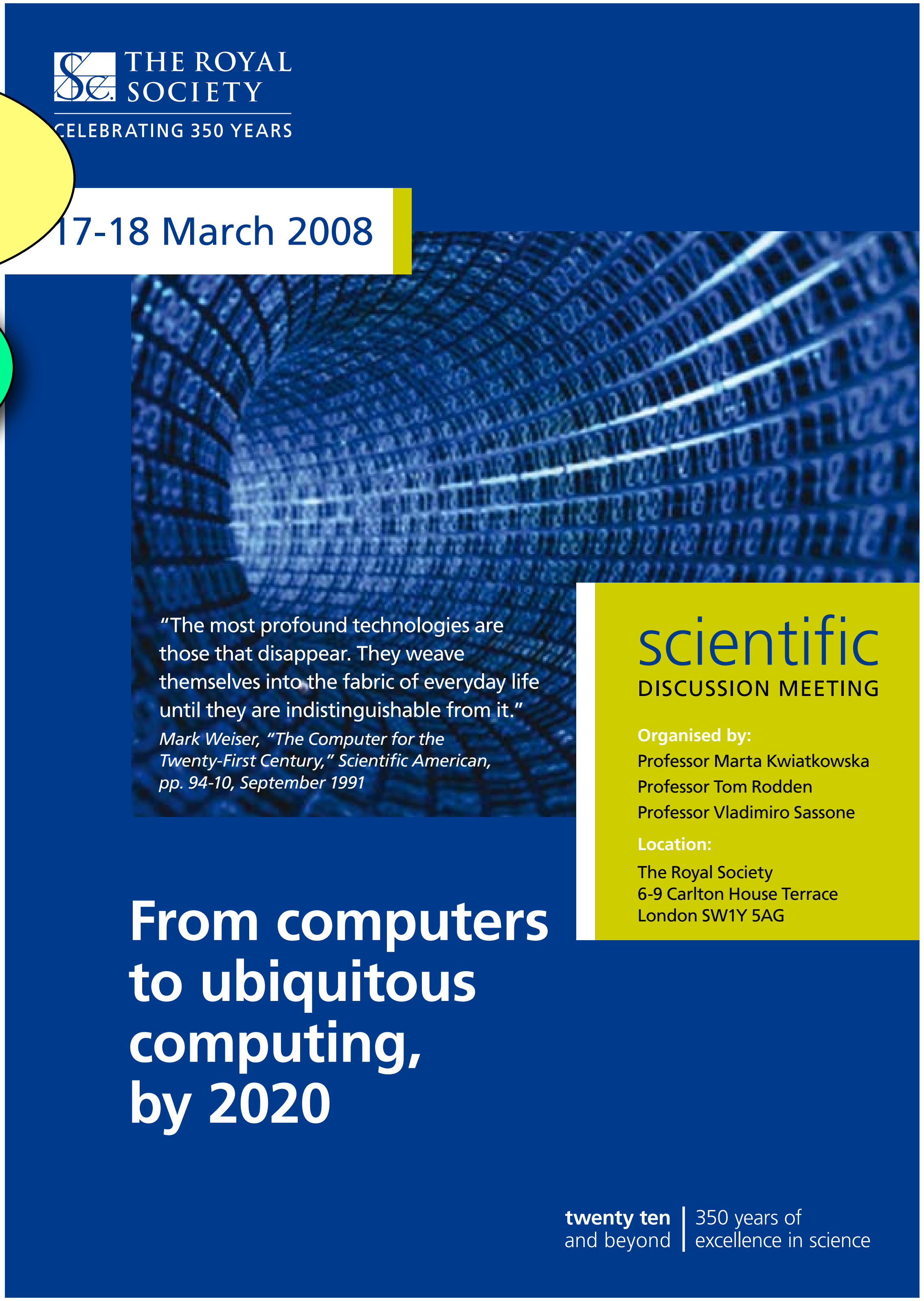
mobile computing

autonomic computing



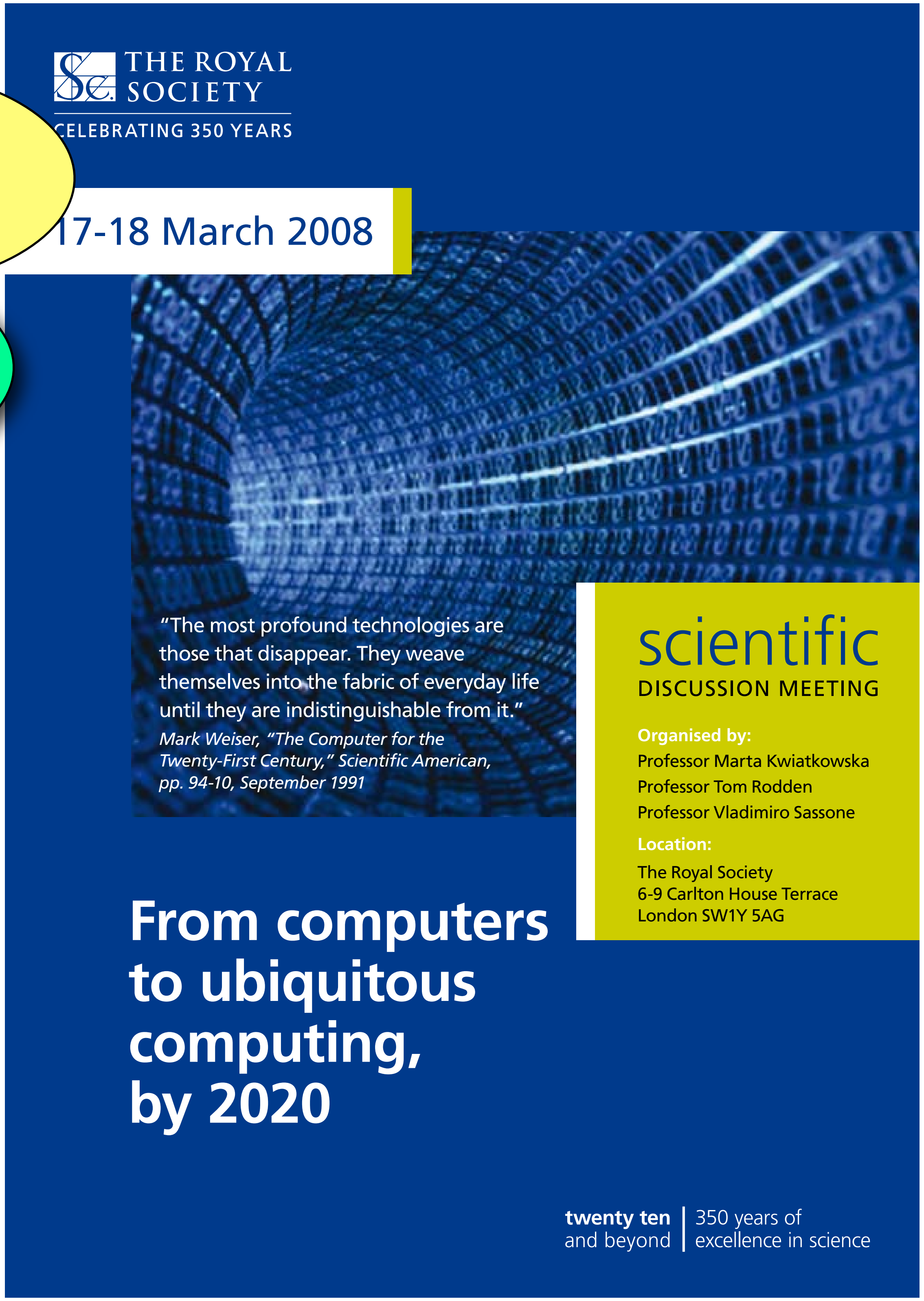
Ubiquitous Computing: what's that?

- invisible computing
- networks architectures
- sensor networks
- sentient computing
- progr. languages
- embedded systems
- mobile computing
- autonomic computing



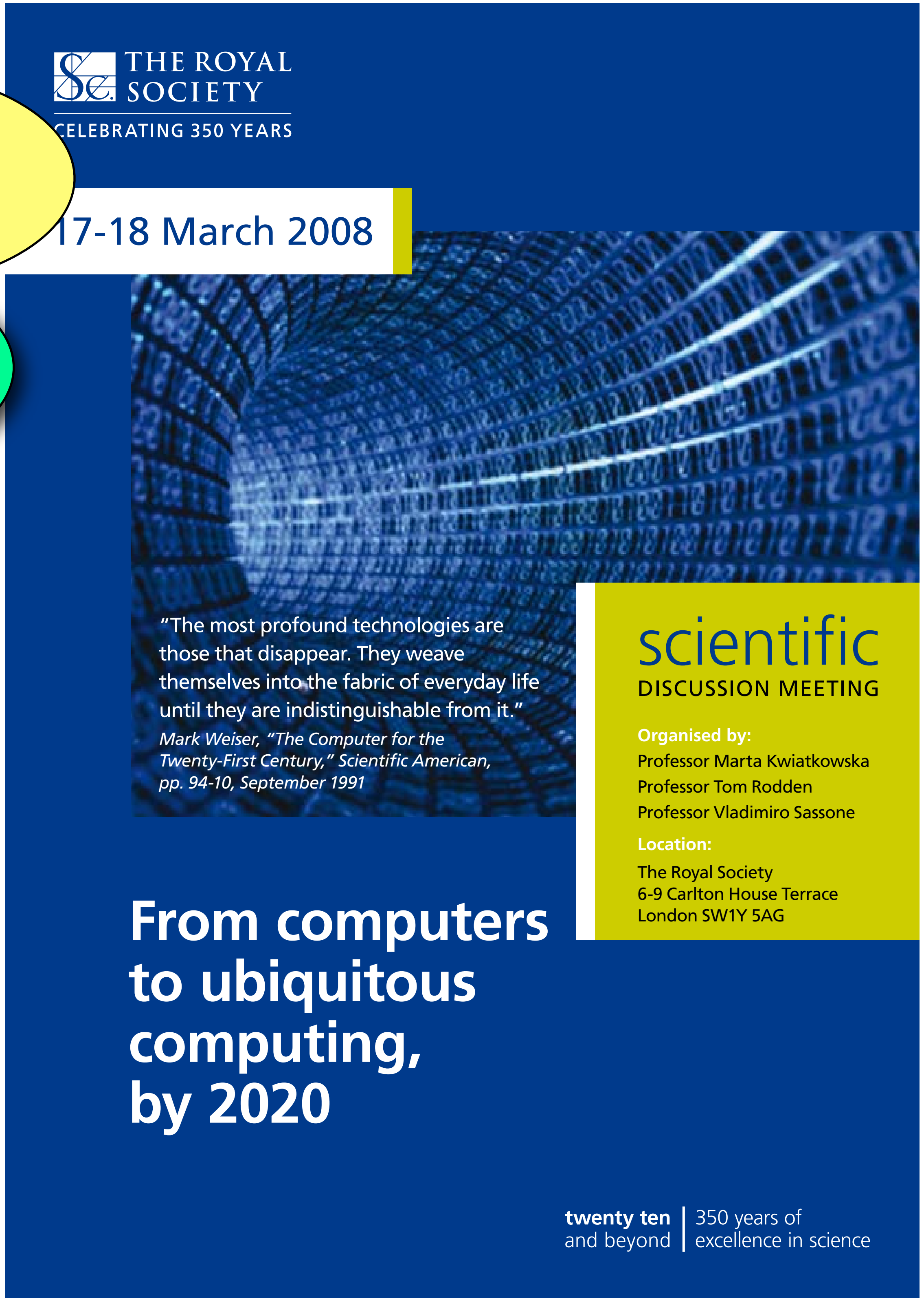
Ubiquitous Computing: what's that?

- invisible computing
- networks architectures
- sensor networks
- sentient computing
- embedded systems
- progr. languages
- mobile computing
- medical computing
- power awareness
- autonomic computing



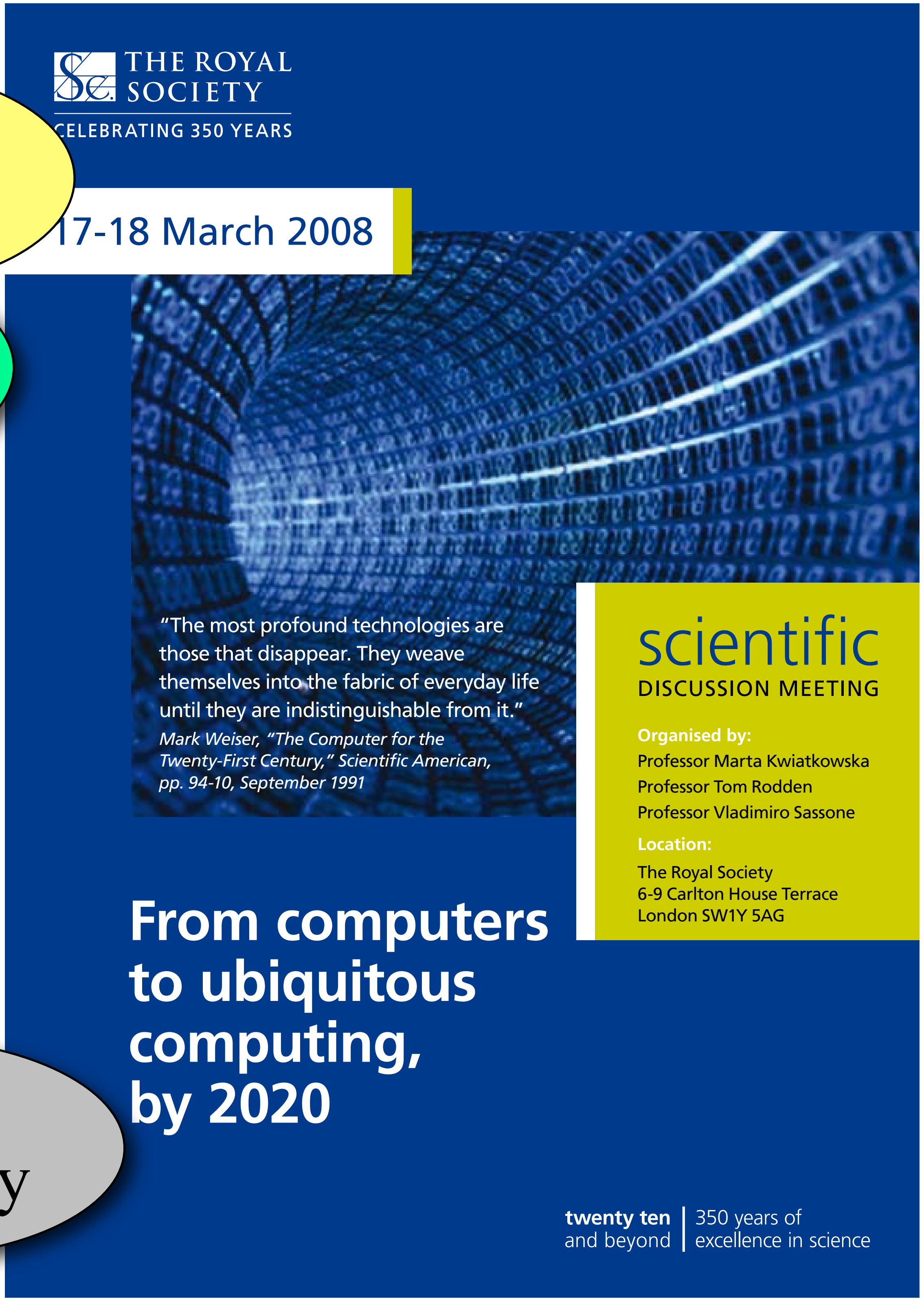
Ubiquitous Computing: what's that?

- invisible computing
- networks architectures
- sensor networks
- sentient computing
- embedded systems
- progr. languages
- mobile computing
- autonomic computing
- medical computing
- power awareness
- legal, social, ethical issues

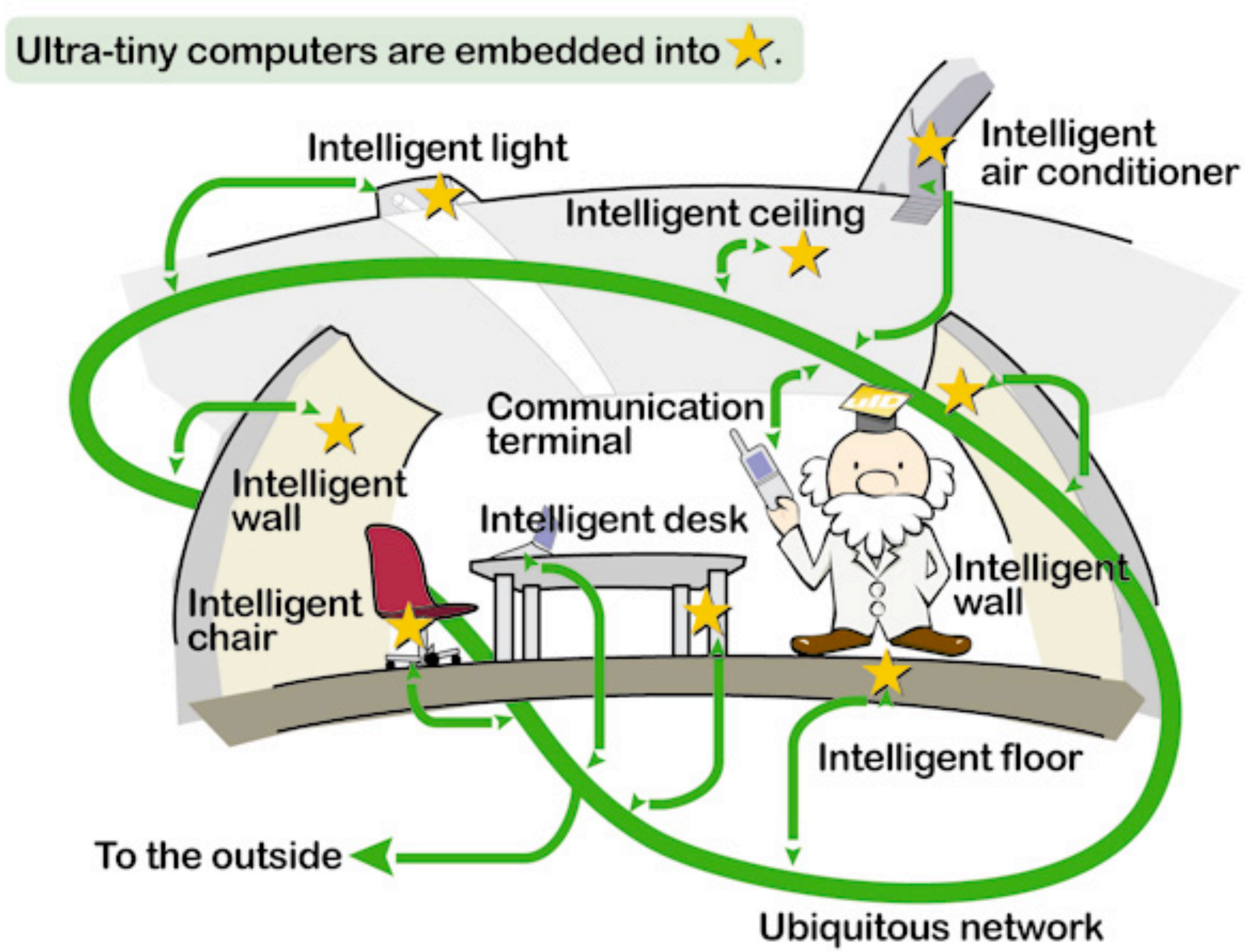


Ubiquitous Computing: what’s that?

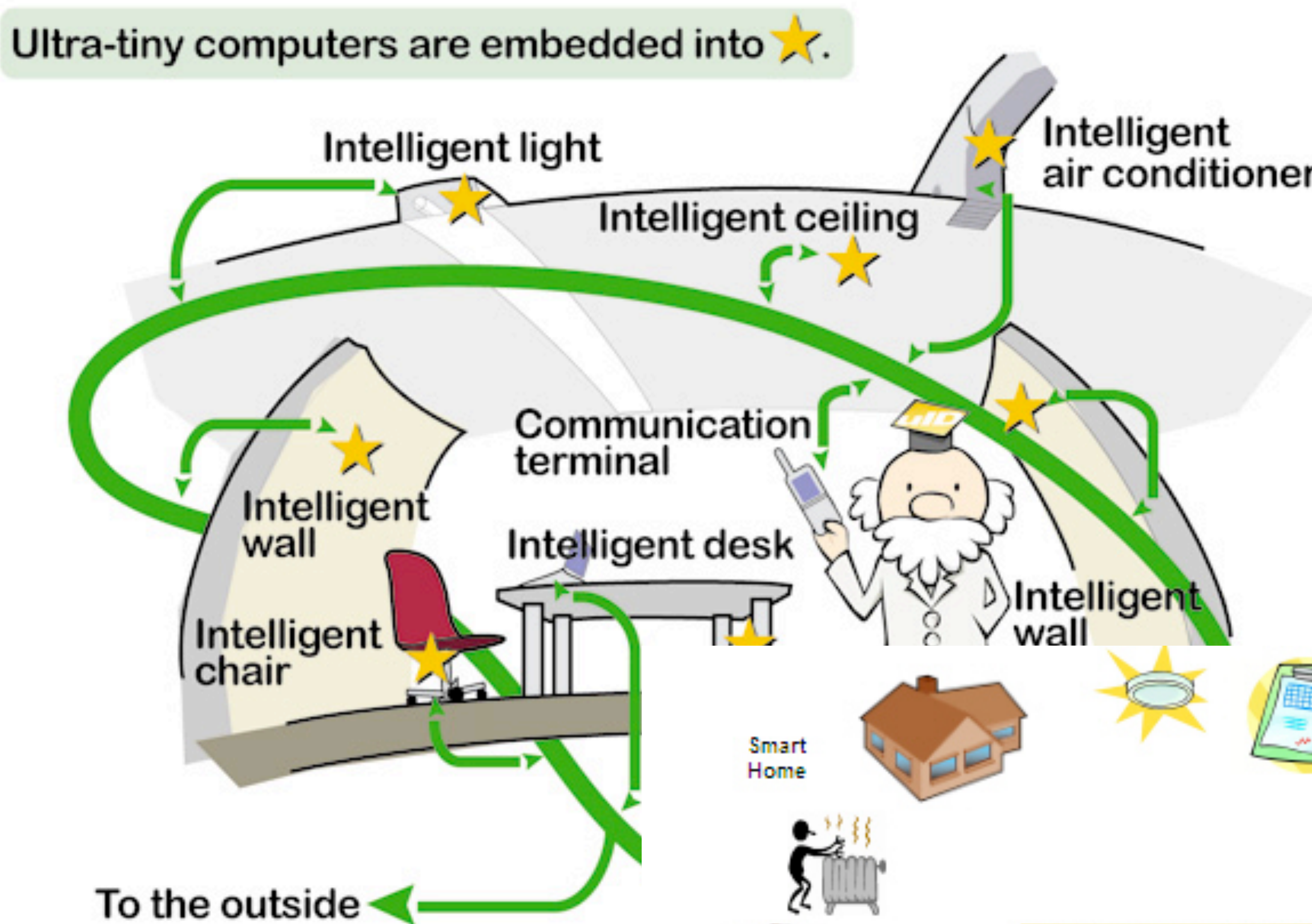
- invisible computing
- networks architectures
- sensor networks
- sentient computing
- embedded systems
- progr. languages
- mobile computing
- medical computing
- power awareness
- autonomic computing
- privacy
- legal, social, ethical issues
- trust
- crypto & security



A lot of embedded devices and smart space



A lot of embedded devices and smart space



Ubiquitous Computing: my perspective

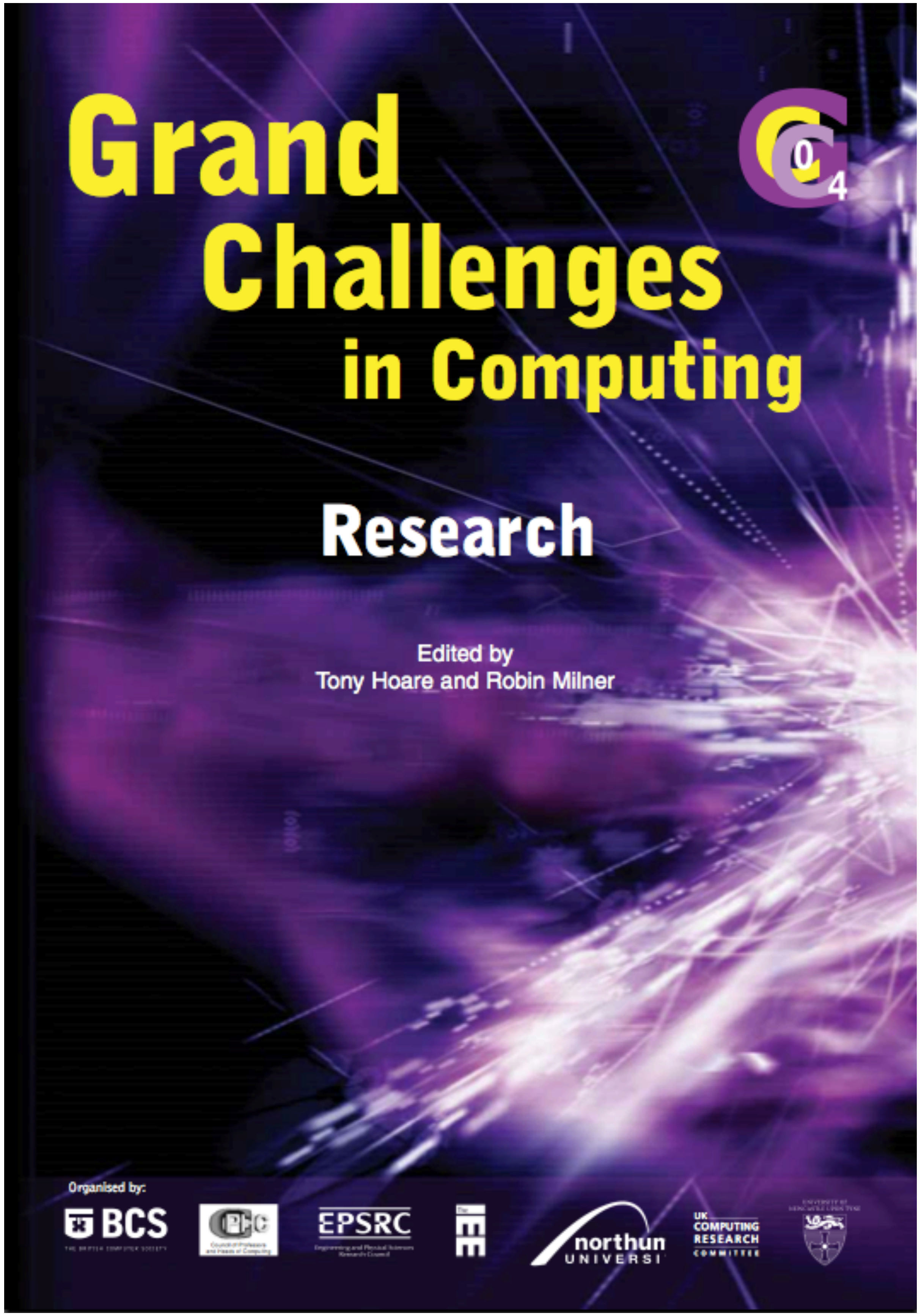
Models for Concurrency

Semantic Theories

Spatial Logics

Programming Languages

Resource Control



Ubiquitous Computing: my perspective

Models for Concurrency

Semantic Theories

Spatial Logics

Programming Languages

Resource Control

Petri Nets Based Models and Calculi

A distributed timed-arc Petri net is a Petri net together with

- a interval time constraint on transitions, either discrete or continuous;
- a clock synchronisation relation Σ on places.

Tokens age, transitions are enabled accordingly. Time elapses at the same speed at p and p' if $p \, \Sigma \, p'$.

Globally Asynchronous, Locally Synchronous

Global Time: $\Sigma = P \times P$ Local Time: $\Sigma = \Delta_P$

A Separation Result: Reachability for safe LT nets is decidable, but undecidable for safe GT nets.

Ubiquitous Computing: my perspective

Models for Concurrency

Semantic Theories

Spatial Logics

Programming Languages

Resource Control

Labels from Reductions

- A categorical machinery which allows the derivation of LTSs from reduction systems.
- Bisimulation on such LTSs is a congruence, provided a general condition is met.

Coinduction Principle Desiderata:

- Correspondence: $p \searrow q$ iff $p \xrightarrow{\tau} q$
- Correctness: $p \approx q$ implies $p \cong q$
- Completeness: $p \cong q$ implies $p \approx q$

The intuition: $a \xrightarrow{\mathscr{C}} b$ iff $\mathscr{C}[a] \searrow b$

Eg: $a \xrightarrow{-|\bar{a}} \mathbf{0}$, $M \xrightarrow{(\lambda x.-)N} M\{N/x\}$, $\mathbf{K}M \xrightarrow{-N} M$

Ubiquitous Computing: my perspective

Models for Concurrency

Semantic Theories

Spatial Logics

Programming Languages

Resource Control

Two related continuations:

(1) What "barbs" i.e. observations are required to give rise to an observation theory corresponding to the contexts as labels ?

(2) How to generate transition systems out of from SOS specification systems in the case of stochastic transition systems?

Ubiquitous Computing: my perspective

- Models for Concurrency
- Semantic Theories
- Spatial Logics
- Programming Languages
- Resource Control

Structural Bilogics

- Spatial logics: Separation in space

$$\ell_1[\ a@\ell.P\ ] \mid \ell_2[\ \overline{a}@\ell.Q\ ]$$

- Separation logics: Separation of resources

$$\ell[\ a.nil \mid b.nil\ ]$$

A more expressiveness and unified approach: Eg,

$$PC_a(in_c \otimes \mathbf{T}) \overset{c}{\otimes} PC_b(out_c \otimes \mathbf{T})$$

describes two PCs linked to the network by "separated" a and b, and to each other by "shared" c.

Results: Proof and model theory for BiLog, encoding of previous logics, decidability issues.

Ubiquitous Computing: my perspective

- Models for Concurrency
- Semantic Theories
- Spatial Logics
- **Programming Languages**
- Resource Control

Jeeg: concurrent OO with history-sensitive access control

- Java (no synchronized(), wait(), notify(), notifyAll()) for business code;
- Linear Time Temporal Logic for synchronisation code (method guards).

```
public class MyClass {
    sync { m : ϕ; ... }
    ...  //Standard Java class def
}
```

where m is a method identifier and ϕ is an LTL formula. When m is invoked, the thread is holds unless ϕ. When the condition is true, all waiting threads are awaken. m is implicitly synchronised.

Ubiquitous Computing: my perspective

Models for Concurrency

Semantic Theories

Spatial Logics

Programming Languages

Resource Control

Resources: Models, Types, Logics, Languages

- Access Control
- Access Authorisation
- Secrecy for Mobile Agents
- Trust Management
- Bounds Control

Ubiquitous Computing: my perspective

- Models for Concurrency
- Semantic Theories
- Spatial Logics
- Programming Languages
- Resource Control

Resources: Models, Types, Logics, Languages

- Access Control
- Access Authorisation
- Secrecy for Mobile Agents
- Trust Management
- Bounds Control

Trust in UbiCom

UNIVERSITY OF Southampton
Electronics and Computer Science

- Features of Ubiquitous Computing like *scalability*, *mobility*, and *incomplete information* deeply affect security requirements.

- One of the proposed approaches is to use a notion of *computational trust*, resembling the concept of trust among human beings.

Approaches to Trust

- **Credential-based Models**
  - trust predicated on possession of predefined credential
  - eg, password, RSA key, certificate, role, history, provenance, ...
- **Predictive Models ("observe & learn")**
  - a probabilistic model assigns a degree of confidence to a principal's ability to predict another principal's behaviour.
  - eg, the behaviour of a principal *A* may be defined as the probability that interaction with *A* yields a certain outcome.
- Overarching notion: **Trust Policy** express complex conditions based on elementary trust values.

Data Provenance

UNIVERSITY OF Southampton
Electronics and Computer Science

- (Meta)data is almost entirely neglected in the process calculi
- Track data provenance both for its important applications and as an challenging exercise in modelling (meta)data.
  Aim at simplicity:
  - data annotations representing provenance
  - structure, interpretation and management of provenance information
  - provenance tracking
- Provenance-based security (*trust + data confidentiality*)
  - Example: conference submission
- The overall ambition is to underpin and develop practical stuff, like trust-policy languages and protocols, and provenance-middleware

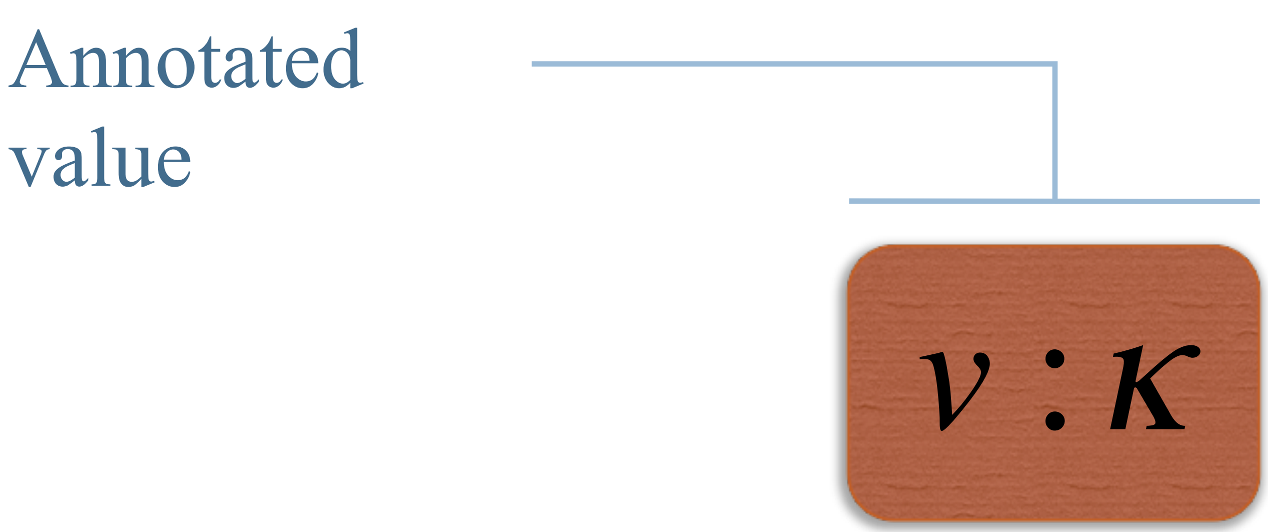
Provenance model

UNIVERSITY OF Southampton
Electronics and Computer Science

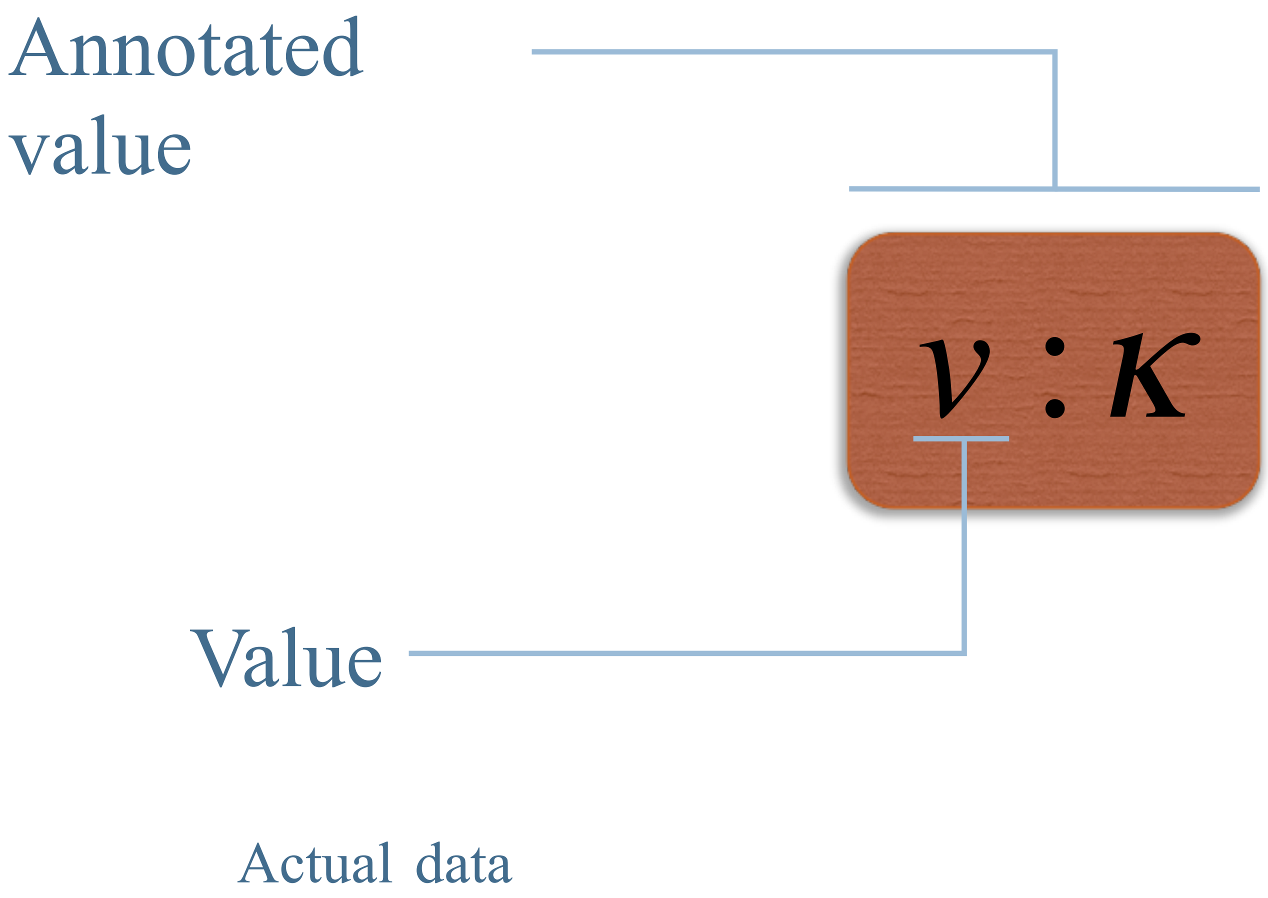
$v : \kappa$

Provenance model

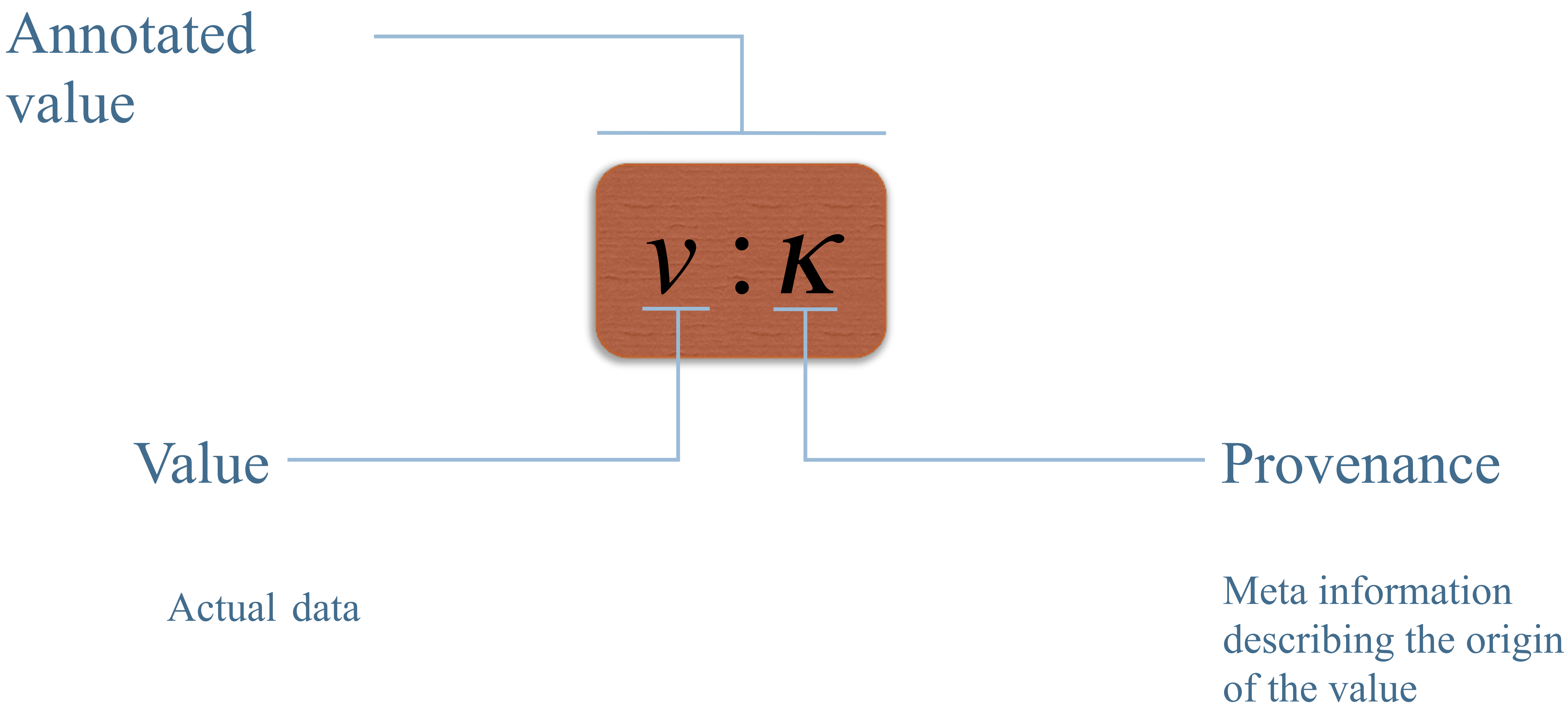
UNIVERSITY OF Southampton
Electronics and Computer Science



Provenance model



Provenance model



Provenance model

Structure and interpretation of provenance

UNIVERSITY OF Southampton
Electronics and Computer Science

$$v : \varepsilon \; ; a!\kappa_1 \; ; b?\kappa_2 \; ; b!(\varepsilon ; c!\kappa_3 , b?\kappa_4) \; ; \ldots$$

Provenance model

Structure and interpretation of provenance

UNIVERSITY OF Southampton
Electronics and Computer Science

Value — Provenance

$$v : \varepsilon\ ; a!\kappa_1\ ; b?\kappa_2\ ; b!(\varepsilon ; c!\kappa_3 , b?\kappa_4)\ ; \ldots$$

Provenance model

Structure and interpretation of provenance

UNIVERSITY OF Southampton
Electronics and Computer Science

Value

Provenance

$$v : \varepsilon ; a!\kappa_1 ; b?\kappa_2 ; b!(\varepsilon ; c!\kappa_3 , b?\kappa_4) ; \ldots$$

“Operations” that were performed on the value. They record the principals that “influenced” the value and how.

Provenance model

Structure and interpretation of provenance

UNIVERSITY OF Southampton
Electronics and Computer Science

ε (empty provenance)
denotes value v originated here

$$v : \varepsilon$$

Provenance model

Structure and interpretation of provenance

UNIVERSITY OF Southampton
Electronics and Computer Science

ε (empty provenance) denotes value *v* originated here

$$v : \varepsilon ; a!\kappa_1$$

It was sent by *a* on a channel with provenance κ_1

Provenance model

Structure and interpretation of provenance

UNIVERSITY OF Southampton
Electronics and Computer Science

ε (empty provenance) denotes value v originated here

$$v : \varepsilon ; a!\kappa_1 ; b?\kappa_2$$

It was sent by a on a channel with provenance κ_1

Was then received by b on a channel with provenance κ_2

Provenance model

Structure and interpretation of provenance

UNIVERSITY OF Southampton
Electronics and Computer Science

ε (empty provenance) denotes value *v* originated here

And then sent by *b* on a channel that *b* received from *c*…

$$v : \varepsilon\ ; a!\kappa_1\ ; b?\kappa_2\ ; b!(\varepsilon ; c!\kappa_3, b?\kappa_4)\ ; \ldots$$

It was sent by *a* on a channel with provenance κ_1

Was then received by *b* on a channel with provenance κ_2

Confidentiality in provenance systems

UNIVERSITY OF Southampton
Electronics and Computer Science

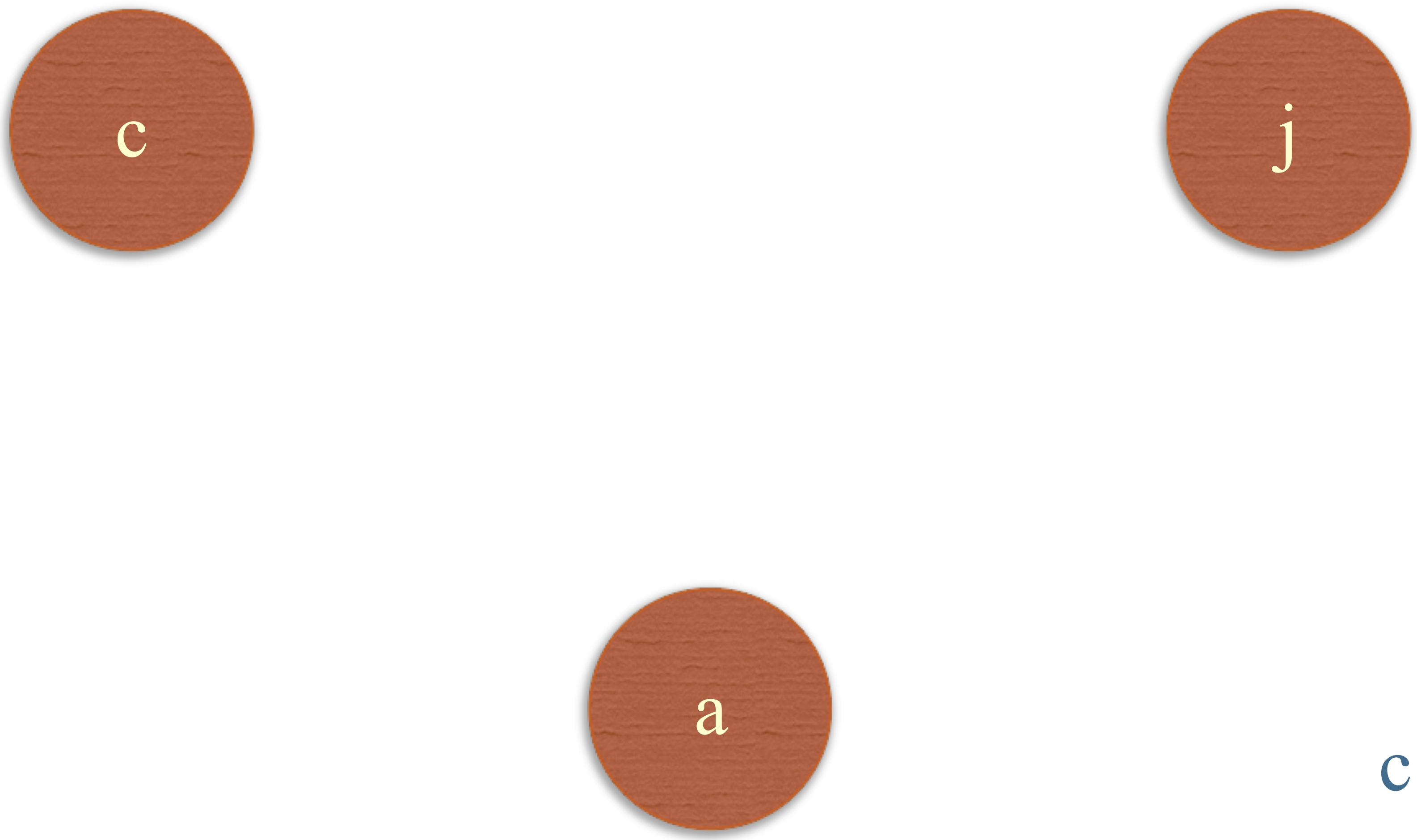
- Data may be public, yet its provenance confidential, or vice versa
- Principals who may access data are not necessarily the same as those who may access its provenance
- Fine grained access control over provenance "histories" is needed as different parts of it have different sensitivity

Security requirements of data $\neq$ Security requirements of its provenance

Hiding provenance trees

Example: conference submissions

UNIVERSITY OF Southampton
Electronics and Computer Science

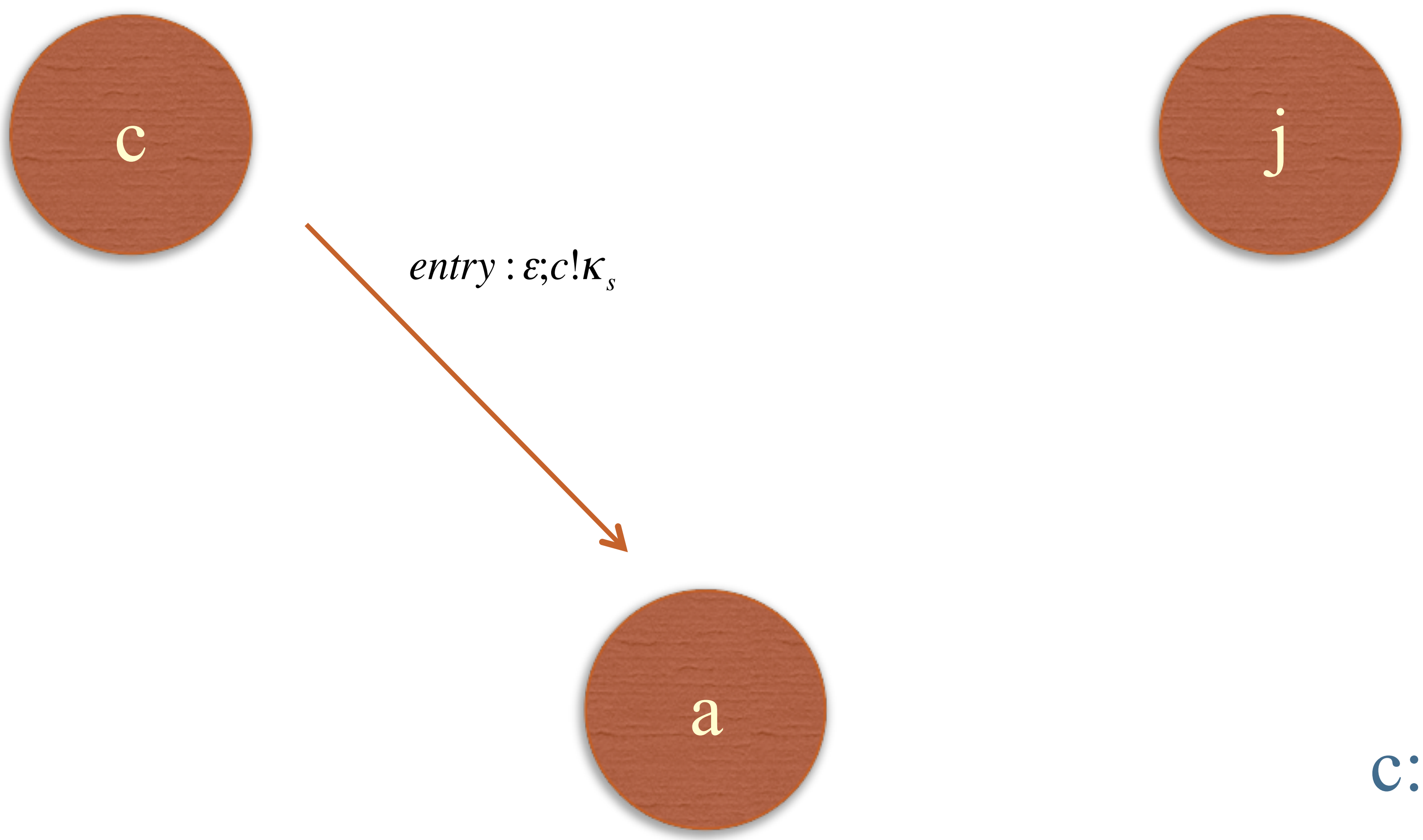


c: Author
a: PC Chair
j: Referee

Hiding provenance trees

Example: conference submissions

UNIVERSITY OF Southampton
Electronics and Computer Science

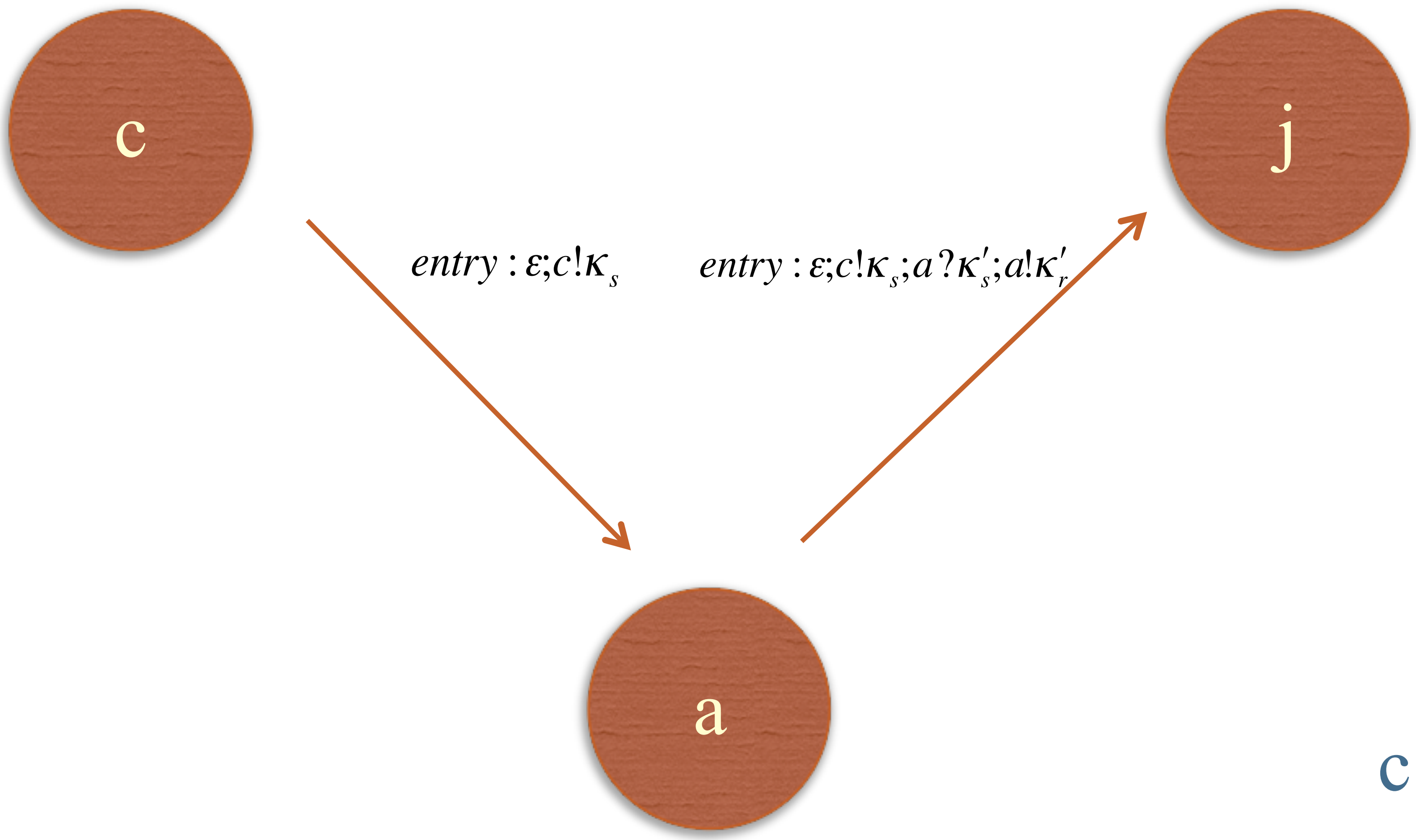


c: Author
a: PC Chair
j: Referee

Hiding provenance trees

Example: conference submissions

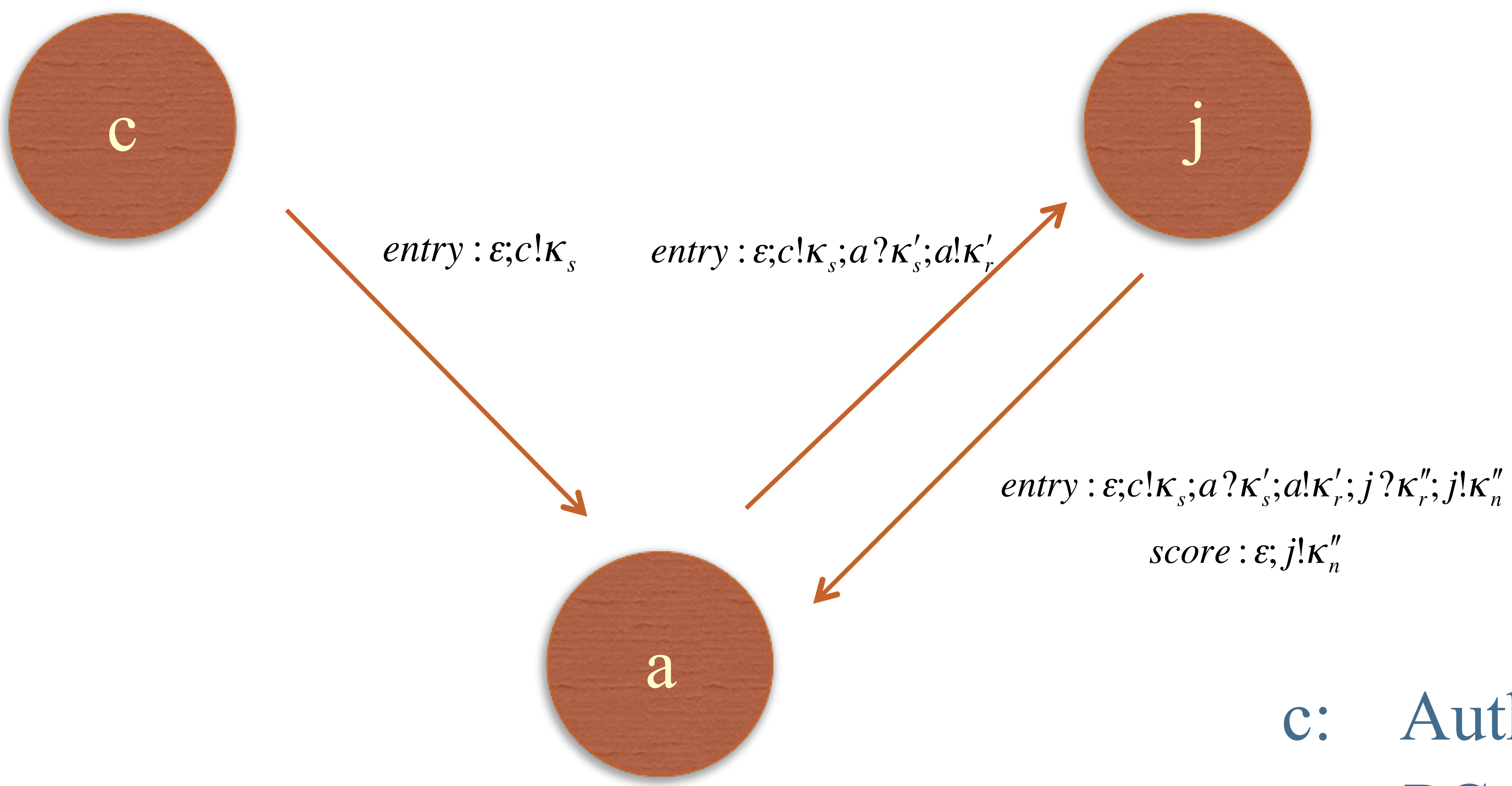
UNIVERSITY OF Southampton
Electronics and Computer Science



c: Author
a: PC Chair
j: Referee

Hiding provenance trees

Example: conference submissions



c: Author

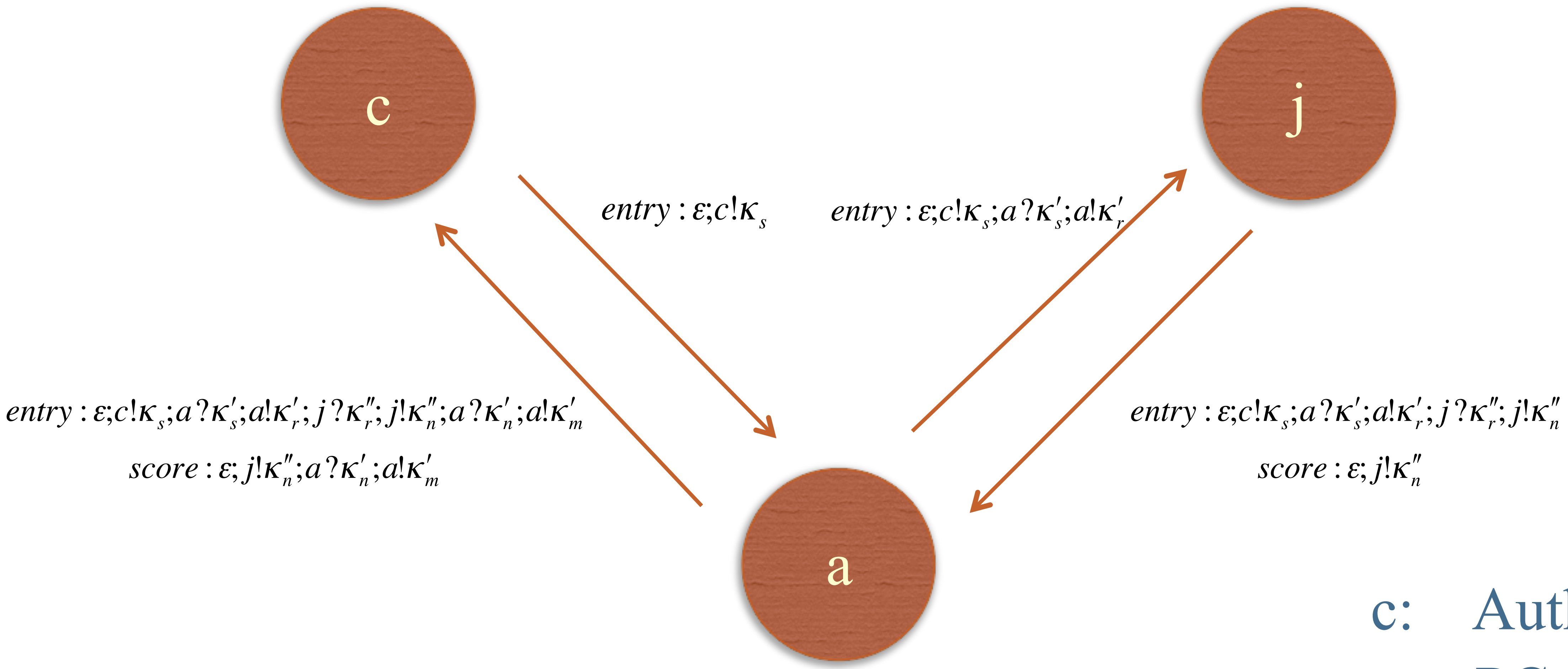
a: PC Chair

j: Referee

Hiding provenance trees

Example: conference submissions

UNIVERSITY OF Southampton
Electronics and Computer Science

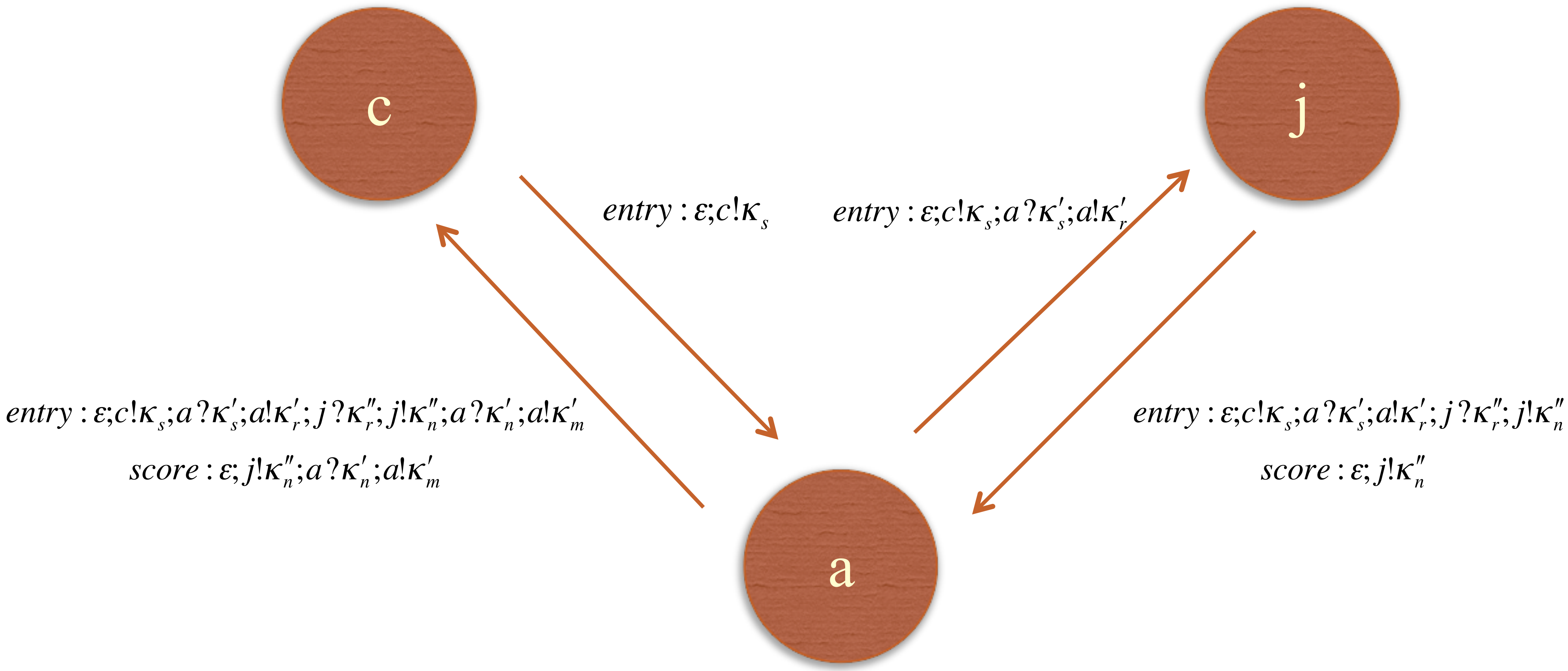


c: Author
a: PC Chair
j: Referee

Hiding provenance trees

Example: conference submissions

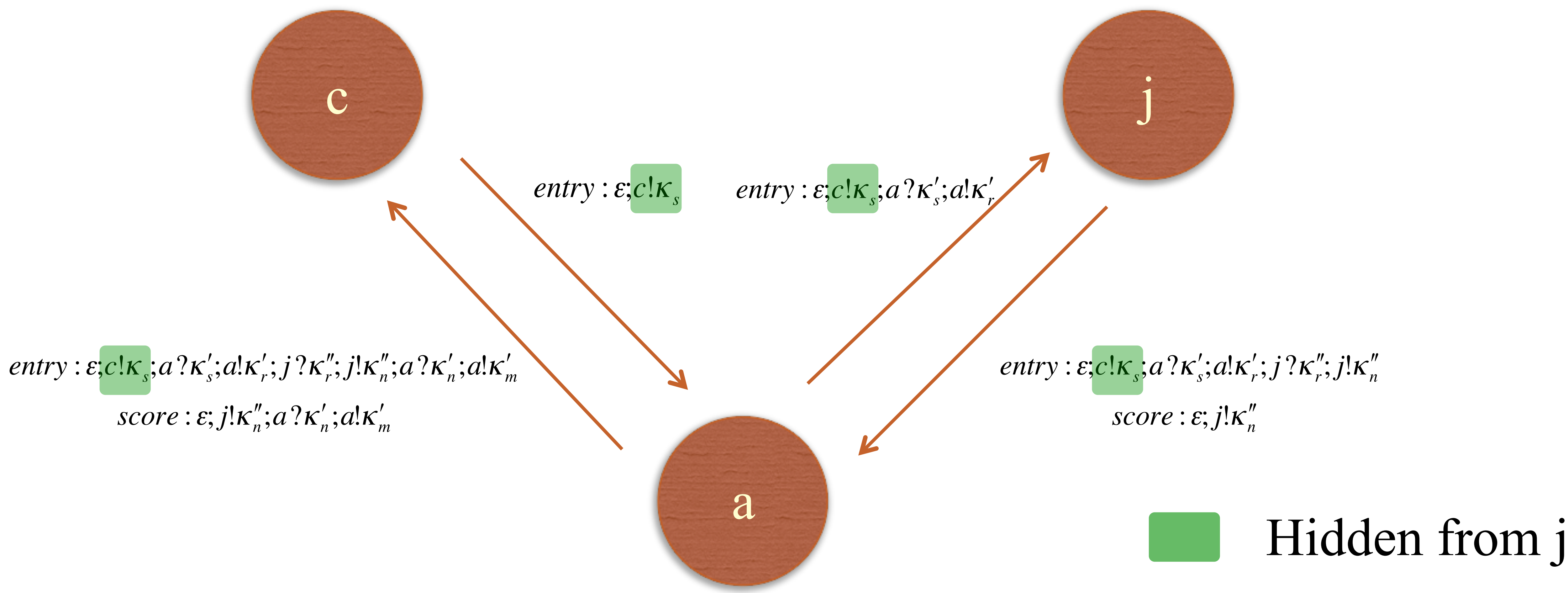
UNIVERSITY OF Southampton
Electronics and Computer Science



Hiding provenance trees

Example: conference submissions

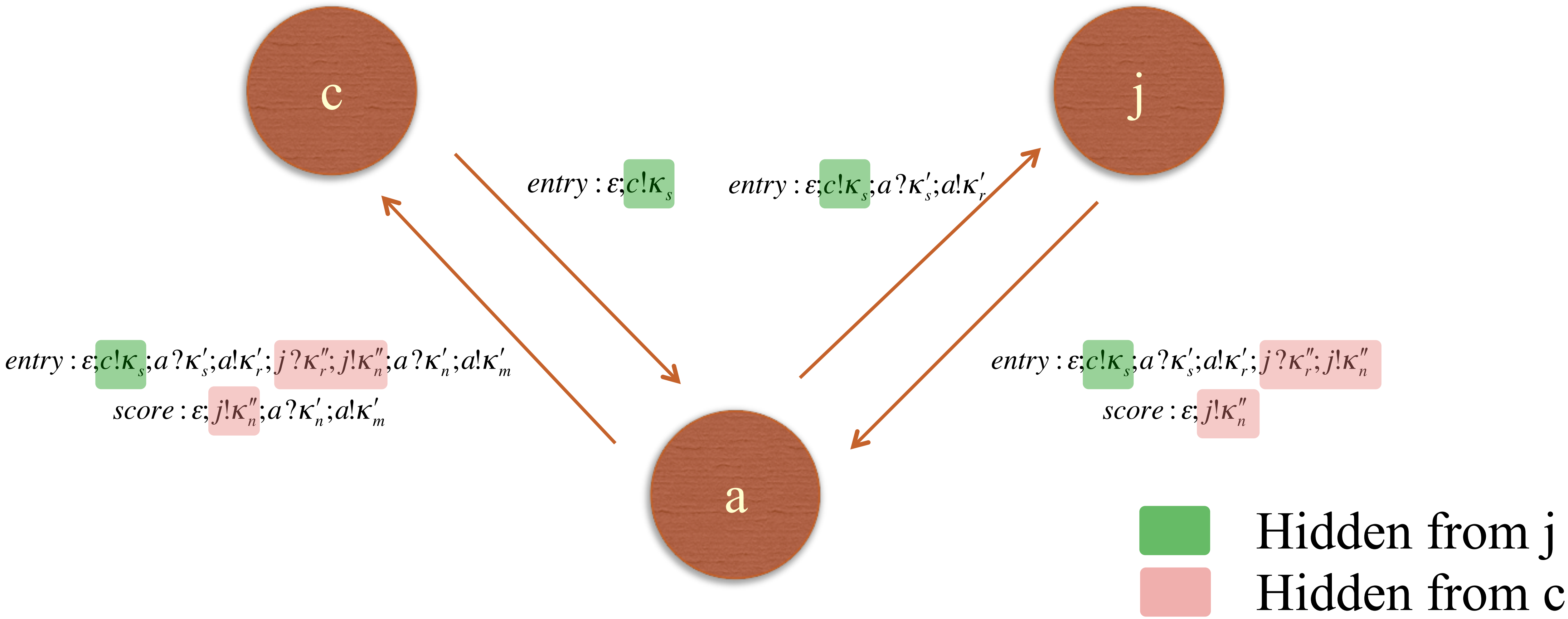
UNIVERSITY OF Southampton
Electronics and Computer Science



Hiding provenance trees

Example: conference submissions

UNIVERSITY OF Southampton
Electronics and Computer Science



Multiple provenance views

UNIVERSITY OF Southampton
Electronics and Computer Science

- One value, multiple **views**

  Different principals have different views of the same provenance list based on their privileges

$$entry : \varepsilon; c!\kappa_s; a?\kappa'_s; a!\kappa'_r; j?\kappa''_r; j!\kappa''_n; a?\kappa'_n; a!\kappa'_m$$

Multiple provenance views

UNIVERSITY OF Southampton
Electronics and Computer Science

- One value, multiple **views**

  Different principals have different views of the same provenance list based on their privileges

$$entry : \varepsilon; c!\kappa_s; a?\kappa'_s; a!\kappa'_r; j?\kappa''_r; j!\kappa''_n; a?\kappa'_n; a!\kappa'_m$$

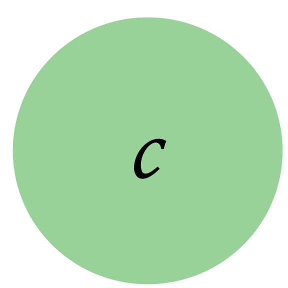
a

Multiple provenance views

- One value, multiple **views**

  Different principals have different views of the same provenance list based on their privileges

$$entry : \varepsilon; c!\kappa_s; a?\kappa'_s; a!\kappa'_r; j?\kappa''_r; j!\kappa''_n; a?\kappa'_n; a!\kappa'_m$$



Multiple provenance views

UNIVERSITY OF
Southampton
Electronics and
Computer Science

- One value, multiple **views**

  Different principals have different views of the same provenance list based on their privileges

$$entry : \varepsilon; c!\kappa_s; a?\kappa'_s; a!\kappa'_r; j?\kappa''_r; j!\kappa''_n; a?\kappa'_n; a!\kappa'_m$$

j

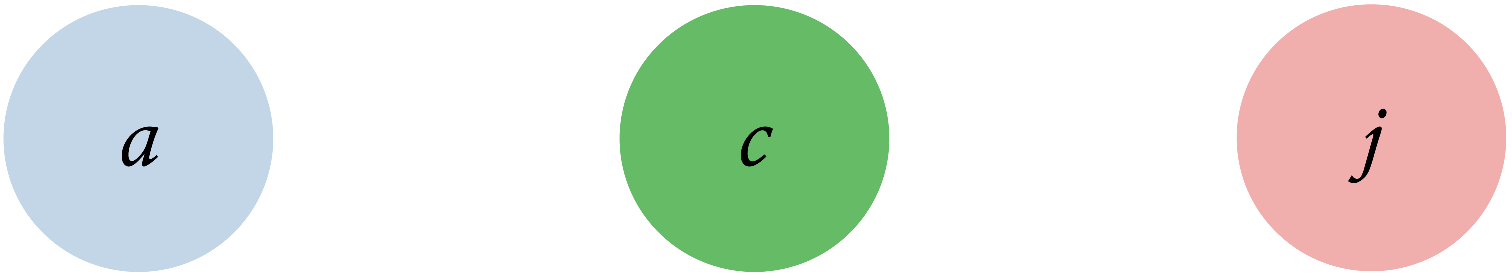
Multiple provenance views

UNIVERSITY OF Southampton
Electronics and Computer Science

- One value, multiple **views**

  Different principals have different views of the same provenance list based on their privileges

$$entry : \varepsilon; c!\kappa_s; a?\kappa'_s; a!\kappa'_r; j?\kappa''_r; j!\kappa''_n; a?\kappa'_n; a!\kappa'_m$$



Inferring probability distributions

- Examples of applications in trust & security
  - Estimate trust in an individual or set of individuals
  - Estimate input distribution of a noisy channel to compute the Bayes risk
  - Apply the Bayesian approach to hypothesis testing (anonymity, information flow)
  - ...

Beta Trust Model

UNIVERSITY OF Southampton
Electronics and Computer Science

- The outcome of an interaction between a principal a *and* a partner b is either *successful* or *unsuccessful*:

$$o \in \{Succ, \ Fail\}$$

- The probability that a partner b interacts successfully with a is governed by the parameter θ where:

$$\theta = \Pr(o = Succ)$$

- Goal: infer (an approximation of) the probability of success
- Means: Observe sequence of trials (observations)

Beta Trust Model

UNIVERSITY OF Southampton
Electronics and Computer Science

- Note that: the behaviour of the partner b represented by θ is assumed to be fixed over time.
- The estimated probability of success, $B(Succ\,|o)$, at time t is the expected value of θ *given the sequence of outcomes*

$$o = \{o_0, o_1, \ldots, o_t\}$$

$$B(\,Succ \mid o) = E[\,\theta \mid o\,]$$

Using evidence to infer θ

- The "*Frequentist*" method:

$$F(n, s) = \frac{s}{n}$$

- The "*Bayesian*" method:

Assume an *a priori* probability distribution for θ (representing your partial knowledge about θ, whatever the source may be) and combine it with the *evidence*, using Bayes' theorem, to obtain the *a posteriori* distribution

A Bayesian approach

UNIVERSITY OF Southampton
Electronics and
Computer Science

- Assumption: θ is the generic value of a continuous random variable Θ whose probability density is a *Beta distribution* with (unknown) parameters σ , φ

$$B(\sigma,\varphi)(\theta) = \frac{\Gamma(\sigma+\varphi)}{\Gamma(\sigma)\Gamma(\varphi)} \; \theta^{\sigma-1}(1-\theta)^{\varphi-1}$$

where Γ is the extension of the factorial function
i.e. $\Gamma(n) = (n-1)!$ for n natural number

- The uniform distribution is a particular case of Beta, for $\sigma = 1,\ \varphi = 1$

- B(σ , φ) can be seen as the a posteriori probability density of Θ given by a uniform a priori (principle of *maximum entropy*) and a trial sequence resulting in σ -1 successes and φ -1 failures.

The Bayesian Approach

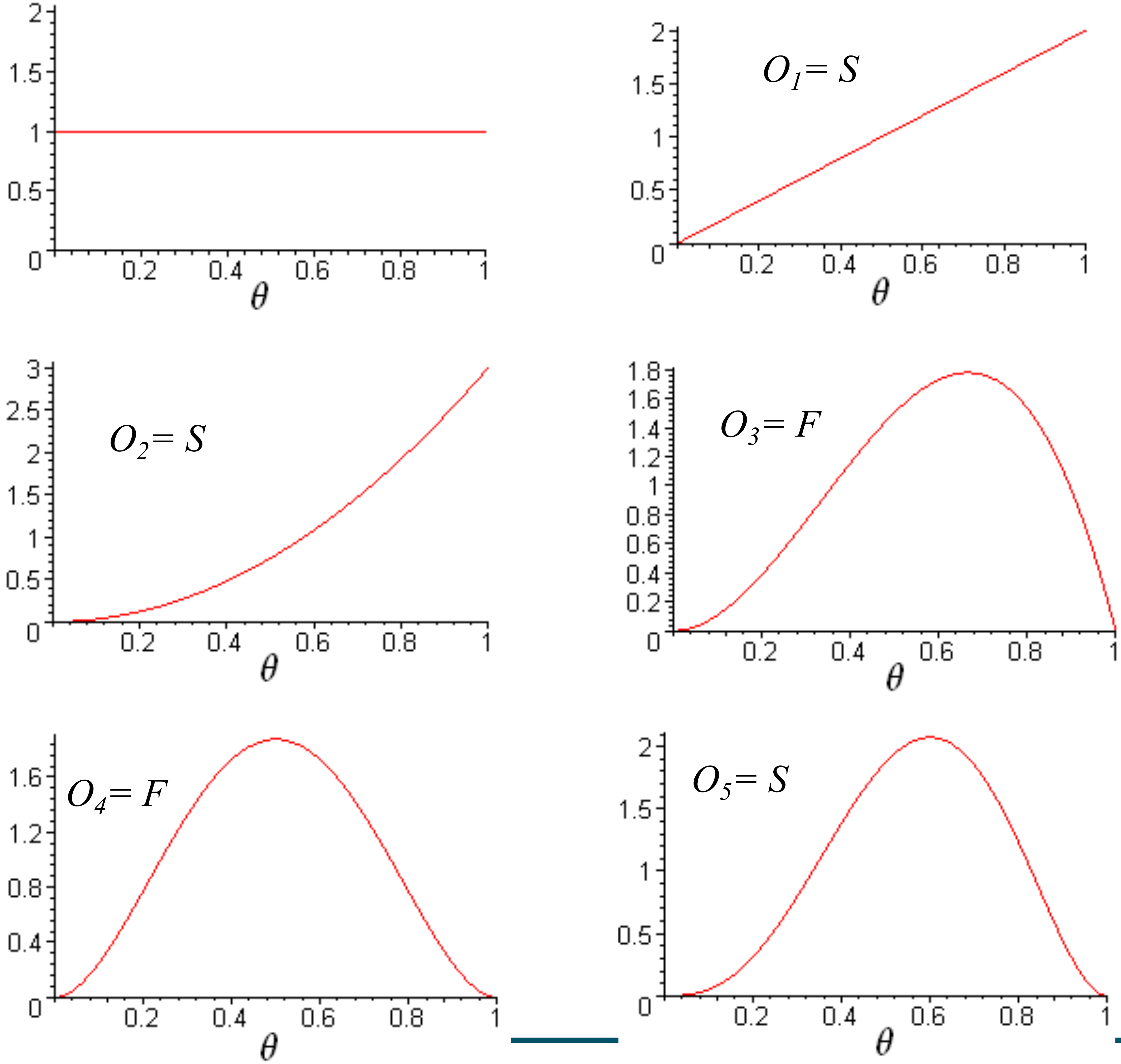
UNIVERSITY OF Southampton
Electronics and Computer Science

- Following the approach, we have three probability density functions for Θ:
  - $B(\sigma, \varphi)$: the "real" distribution of Θ
  - $B(\alpha, \beta)$: the *a priori* (our estimate of the distribution of Θ)
  - $B(s+\alpha, f+\beta)$: the *a posteriori* (the distribution of Θ after the trials)
- The result of the mean-based algorithm is :

$$A_{\alpha,\beta}(n,s) = E_{B(s+\alpha,f+\beta)}(\Theta) = \frac{s+\alpha}{s+f+\alpha+\beta} = \frac{s+\alpha}{n+\alpha+\beta}$$

Trust Inference Process

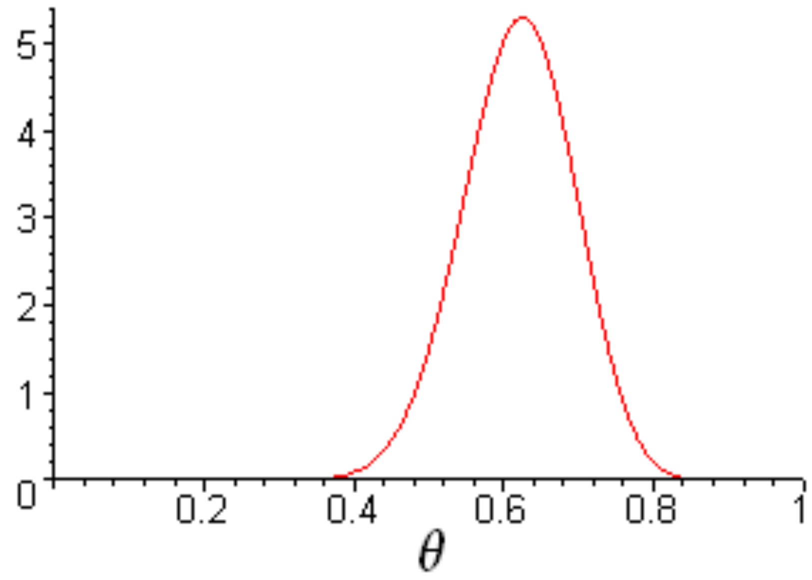
UNIVERSITY OF
Southampton
Electronics and
Computer Science



Trust Inference Process

UNIVERSITY OF Southampton
Electronics and Computer Science

The distribution of θ after 40 interactions
25 Successful and 15 Failed

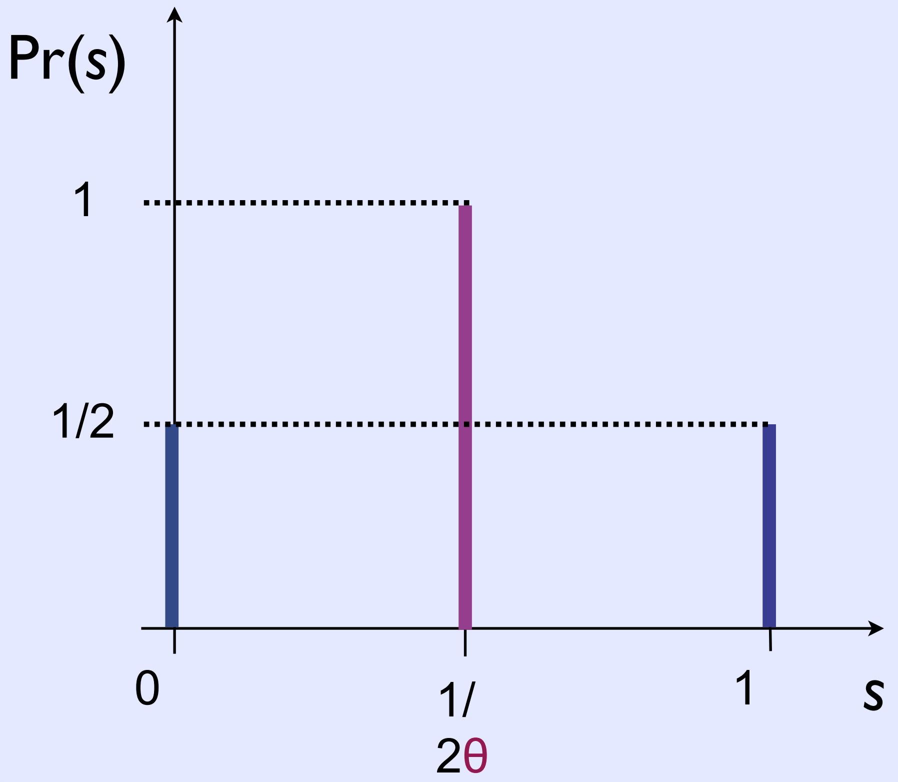


Bayesian vs Frequentist

UNIVERSITY OF Southampton
Electronics and Computer Science

The Frequentist approach can be worse than the Bayesian approach even when the trials give a "good" result, or when we consider the average difference (from the "true" θ) wrt all possible results

Example: "true θ" = 1/2, n = 1



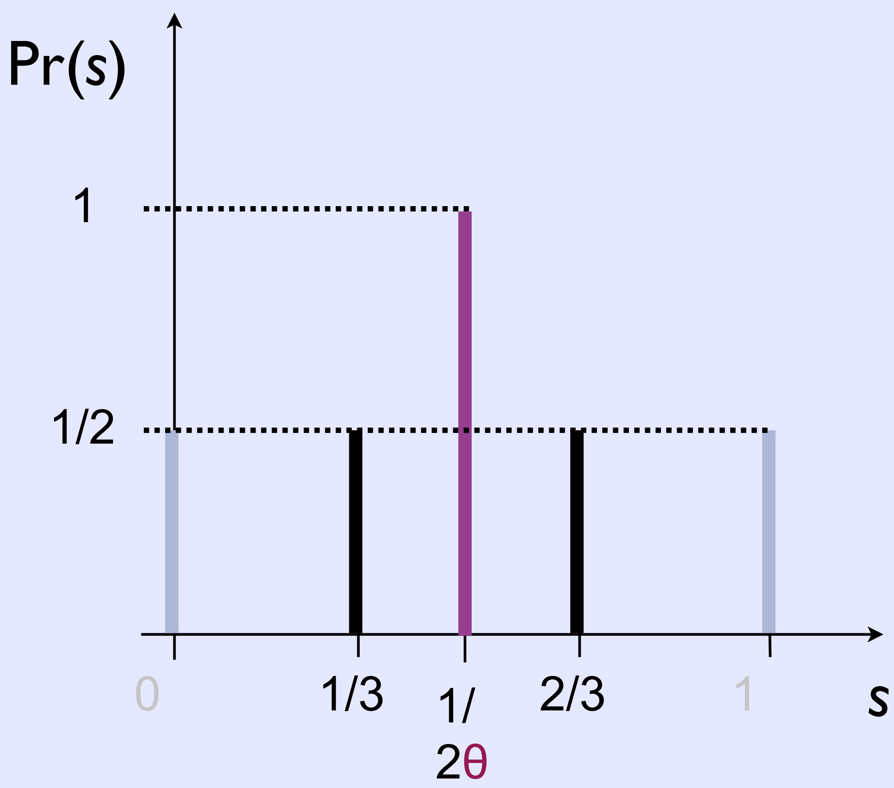
$$F(n,s) = \frac{s}{n} = \begin{cases} 0 & s = 0 \\ 1 & s = 1 \end{cases}$$

The difference from the true distribution is 1/2

Bayesian vs Frequentist

The Frequentist approach can be worse than the Bayesian approach even when the trials give a "good" result, or when we consider the average difference (from the "true" θ) wrt all possible results

Example: "true θ" = 1/2, n = 1



$$F(n,s) = \frac{s}{n} = \begin{cases} 0 & s = 0 \\ 1 & s = 1 \end{cases}$$

The difference from the true distribution is 1/2

A better function would be

$$F_c(n,s) = \frac{s+1}{n+2} = \begin{cases} \frac{1}{3} & s = 0 \\ \frac{2}{3} & s = 1 \end{cases}$$

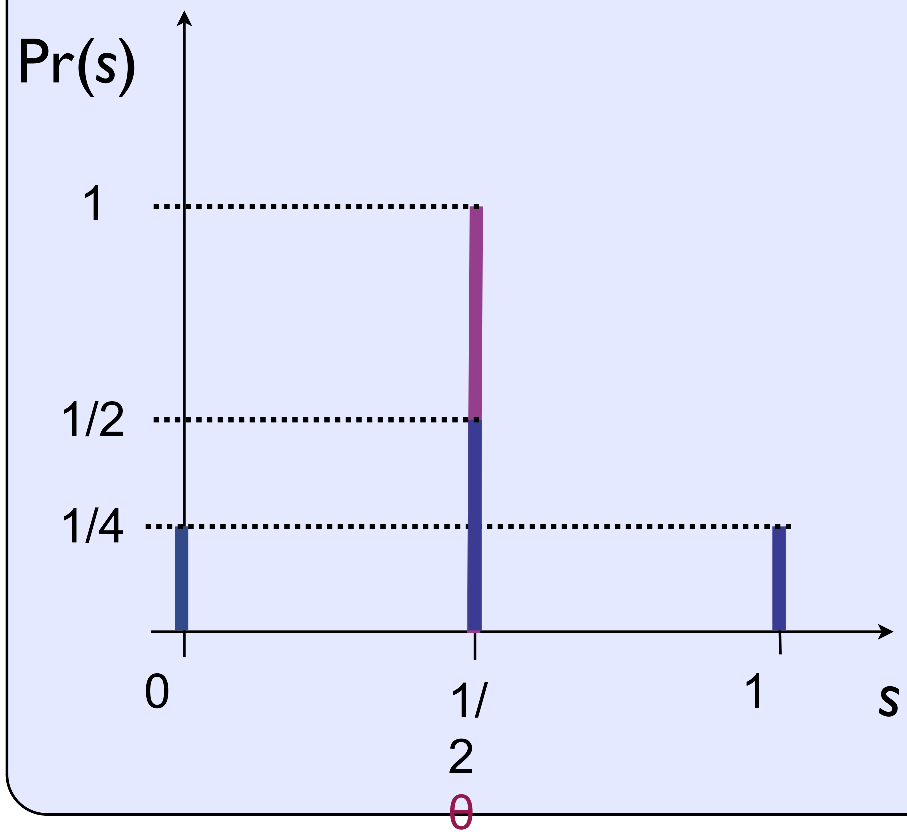
The difference from the true distribution is 1/6

Bayesian vs Frequentist

UNIVERSITY OF Southampton
Electronics and Computer Science

The Frequentist approach can be worse than the Bayesian approach even when the trials give a "good" result, or when we consider the average difference (from the "true" θ) wrt all possible results

Example: "true θ" = 1/2, n = 2



$$F(n,s) = \frac{s}{n} = \begin{cases} 0 & s=0 \\ \frac{1}{2} & s=1 \\ 1 & s=2 \end{cases}$$

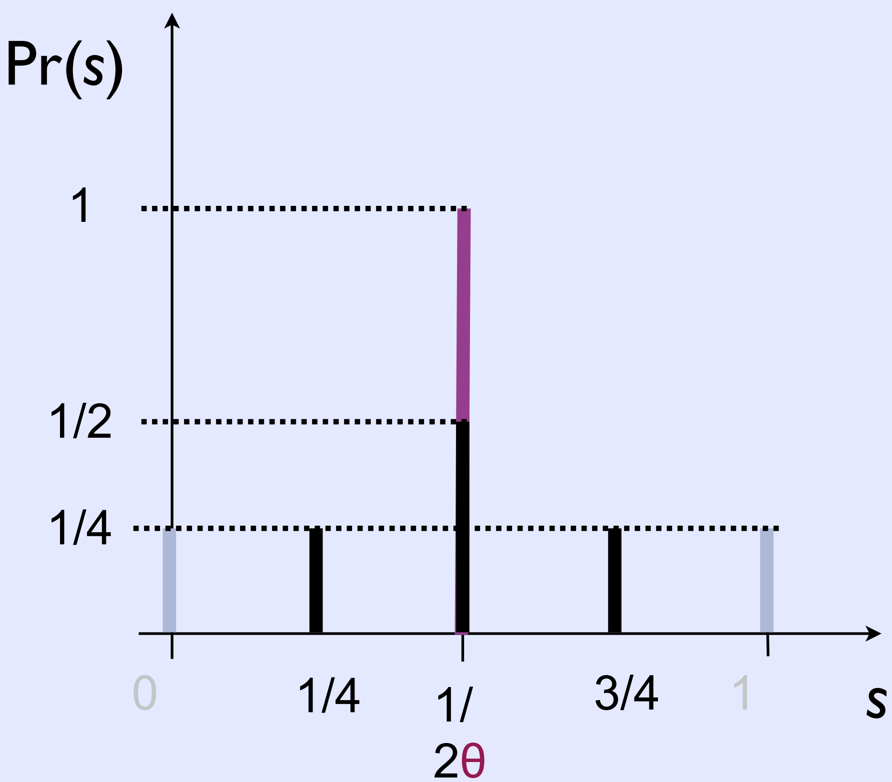
The average difference from the true distribution is 1/4

UNIVERSITY OF Southampton
Electronics and Computer Science

Bayesian vs Frequentist

The Frequentist approach can be worse than the Bayesian approach even when the trials give a "good" result, or when we consider the average difference (from the "true" θ) wrt all possible results

Example: "true θ" = 1/2, n = 2



$$F(n,s) = \frac{s}{n} = \begin{cases} 0 & s=0 \\ \frac{1}{2} & s=1 \\ 1 & s=2 \end{cases}$$

The average distance from the true distribution is 1/4

Again, a better function would be

$$F_c(n,s) = \frac{s+1}{n+2} = \begin{cases} \frac{1}{4} & s=0 \\ \frac{1}{2} & s=1 \\ \frac{3}{4} & s=2 \end{cases}$$

The average distance from the true distribution is 1/8

Measuring the precision of Bayesian algorithms

UNIVERSITY OF Southampton
Electronics and Computer Science

- Define a "difference" $D(A(n,s), \theta)$ (not necessarily a distance)
  - non-negative
  - zero iff $A(n,s) = \theta$

- Consider the expected value $D_E(A,n,\theta)$ of $D(A(n,s), \theta)$ with respect to the likelihood (the conditional probability of $s \mid \theta$)

- *Risk of A* : the expected value $R(A,n)$ of $D_E(A,n,\theta)$ with respect to the "true" distribution of Θ

$$D_E(A,n,\theta) = \sum_{s=0}^{n} Pr(s \mid \theta)\ D(A(n,s),\theta)$$

$$R(A,n) = \int_0^1 Pd(\theta)\ D_E(A,n,\theta)\ d\theta$$

Measuring the precision of Bayesian algorithms

UNIVERSITY OF Southampton
Electronics and Computer Science

We have considered the following candidates for *D*(*x*,*y*) (all of which can be extended to the n-ary case):

- The norms:
  - $|x - y|$
  - $|x - y|^2$
  - …
  - $|x - y|^k$
  - …
- The Kullback-Leibler divergence

$$D_{KL}((y, 1-y) \parallel (x, 1-x)) = y \ \log_2 \frac{y}{x} + (1-y) \log_2 \frac{1-y}{1-x}$$

Measuring the precision of Bayesian algorithms

UNIVERSITY OF Southampton
Electronics and Computer Science

- *Theorem.* For the mean-based Bayesian algorithms, with *a priori* B (α, β), we have that the condition is satisfied (i.e. the Risk is minimum when α, β coincide with the parameters σ, φ of the "true" distribution), by the following functions:

  - The 2nd norm $(x - y)^2$
  - The Kullback-Leibler divergence

- Surprising that the condition is satisfied by these two very different functions, and not by any of the other norms $|x - y|^k$ for $k \neq 2$.

- It leaves the search open for a measure for assessment and comparison of trust algorithm.

Potential applications

UNIVERSITY OF Southampton
Electronics and Computer Science

- We can use D_E to compare two different estimation algorithms; develop a measure of quality for "*decision-making*" algorithms
  - Mean-based vs other ways of selecting a θ
  - Bayesian vs non-Bayesian
  - In more complicated scenarios there may be different Bayesian mean-based algorithms; eg.: noisy channels.

Potential applications (ctd)

UNIVERSITY OF Southampton
Electronics and Computer Science

- D_E induces a metric on distributions. Bayes' equations define transformations on this metric space from the a priori to the a posteriori.
  - Study the properties of such transformations to reveal interesting properties of the corresponding Bayesian methods, independent of the a priori.
- Hypothesis testing (*privacy, anonymity, confidentiality, information flow analysis, input distribution analysis, ...*) :
  - determine (probabilistic) bounds as to what probability-distribution inference algorithm may determine about you, your online activity, your software

Limitation of the Beta model

- The assumption that a principal behaviour is fixed is not always realistic:

- The behaviour of a principal may depend on its internal state which may change over time.

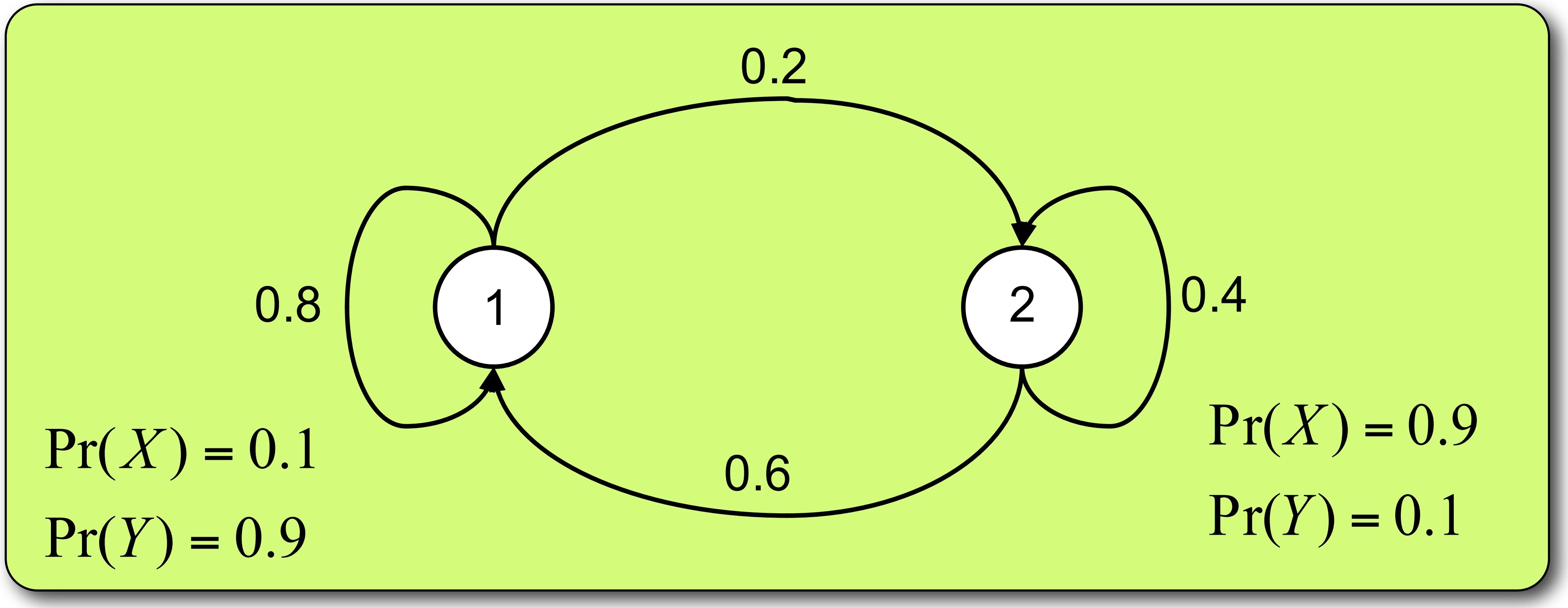
Modelling Dynamic Behaviour

- Modelling static behaviour as a probability distribution over outcomes leads to modelling the dynamic behaviour by a *Hidden Markov Model (HMM).*

- A single state in an HMM models the system behaviour at a particular time.

Hidden Markov Model:

UNIVERSITY OF Southampton
Electronics and
Computer Science



$$S = \{1,2\}$$

$$V = \{X, Y\}$$

$$A = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} \qquad B = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$

A simpler model: Beta with Decay

- The probability distribution over outcomes changes over time.
- Old observations are given less weight (decayed) than more recent observations.
- Weights of observations are controlled by the decay factor r.

Beta Trust Model with Decay

Given a decay factor $0 \leq r < 1$ and an observation sequence $o = \{o_0, \ldots, o_L\}$ then

$$B_r(Succ \mid o) = \frac{m_r(o) + 1}{m_r(o) + n_r(o) + 2} \qquad B_r(Fail \mid o) = \frac{m_r(o) + 1}{m_r(o) + n_r(o) + 2}$$

where

$$m_r(o) = \sum_{i=0}^{L} r^{L-i} \cdot \delta_{Succ}(o_i) \qquad n_r(o) = \sum_{i=0}^{L} r^{L-i} \cdot \delta_{Fail}(o_i)$$

and

$$\delta_x(o) = \begin{cases} 1 & \text{if } x = o \\ 0 & \text{otherwise} \end{cases}$$

How good is the model ?

UNIVERSITY OF
Southampton
Electronics and
Computer Science

- Given a dynamic system modelled by an HMM λ we define Beta estimation error as follows

$$\text{Error}(\lambda, r) = E\left[\,(B(\,Succ \mid o) - \alpha)^2\,\right]$$

where r is the decay factor, and α is the real probability that next outcome is Success

System stability

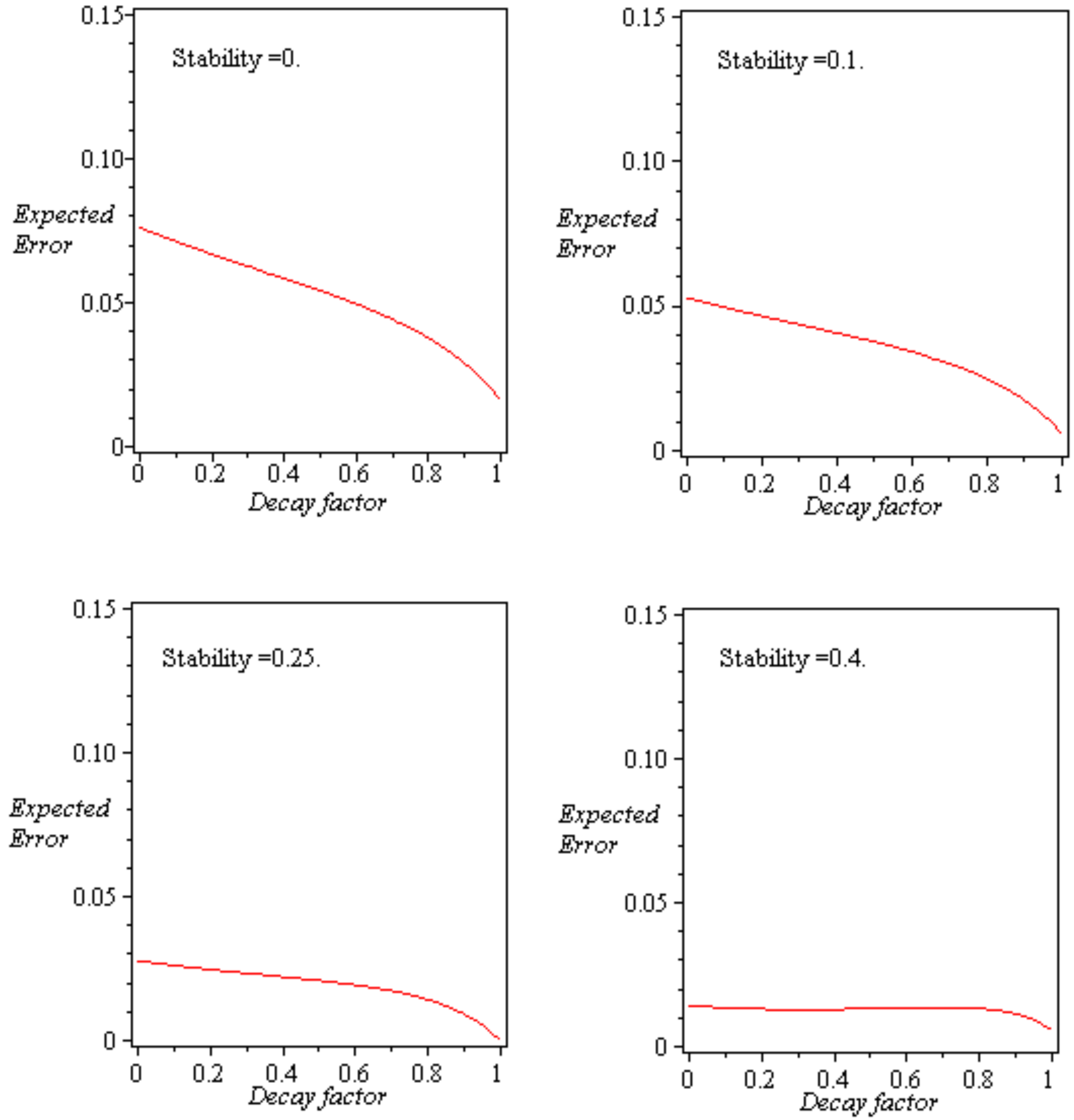
- *System stability* is the expected probability of the HMM remaining in the same state.

- Consider the system modelled by HMM:

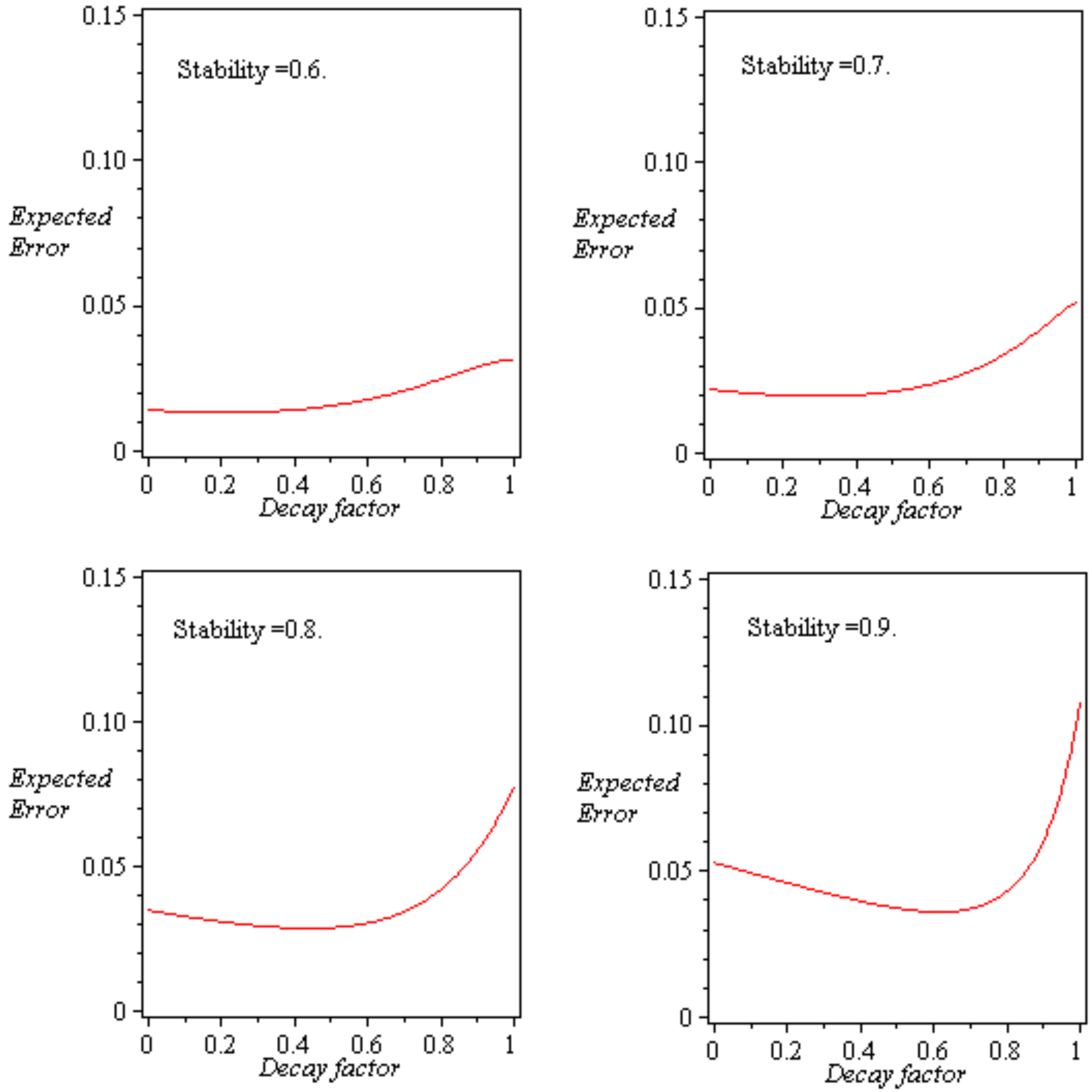
$$A_\lambda = \begin{bmatrix} s & \frac{1-s}{3} & \frac{1-s}{3} & \frac{1-s}{3} \\ \frac{1-s}{3} & s & \frac{1-s}{3} & \frac{1-s}{3} \\ \frac{1-s}{3} & \frac{1-s}{3} & s & \frac{1-s}{3} \\ \frac{1-s}{3} & \frac{1-s}{3} & \frac{1-s}{3} & s \end{bmatrix} \qquad \Theta_\lambda = \begin{bmatrix} 1.0 \\ 0.7 \\ 0.3 \\ 0.0 \end{bmatrix}$$

Unstable system

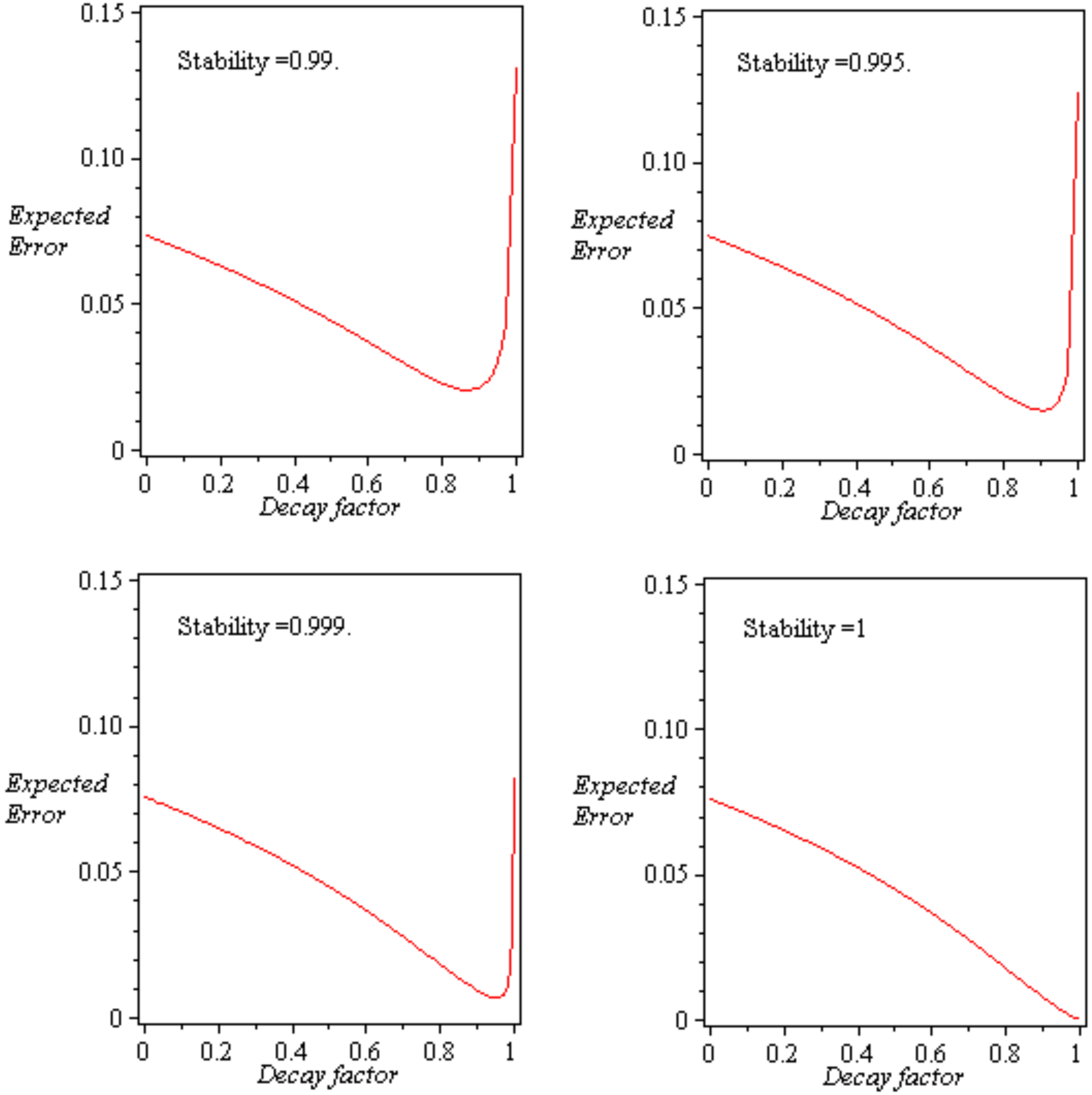
UNIVERSITY OF Southampton
Electronics and Computer Science



Stable system



Very stable system



Conclusion (in general)

UNIVERSITY OF Southampton
Electronics and Computer Science

A whole wholly-different conception of computing to be developed: hard to talk of "further" work in general

Chiefly, nowhere like here apps w/out sound models are dangerous, and theory without practice is pointless

Conclusion (in general)

UNIVERSITY OF Southampton
Electronics and Computer Science

A whole wholly-different conception of computing to be developed: hard to talk of "further" work in general

The gap between *Theory* and *Practice* matters in practice (although it may not matter in theory)

are dangerous, and theory without practice is pointless

Conclusion (in general)

A whole wholly-different conception of computing to be developed: hard to talk of "further" work in general

The gap between *Theory* and *Practice* matters in practice (although it may not matter in theory)

are dangerous, and theory without practice is pointless

One thing I know: as one cannot "*model-check*" UbiNet, *security* & *privacy* in UbiCom must be coupled with *trust*

Conclusion (personal take)

UNIVERSITY OF Southampton
Electronics and Computer Science

in the short term:

- hiding and multiview in provenance trees
- measures suitable to compare trust-algorithms
- reputation in HMMs
- integration of anonymity protocols and trust

in the longer term:

- programming language bindings
- data confidentiality and then privacy
- ...
- ...