UNIK
UNIVERSITY GRADUATE CENTER

Trust and Reputation Systems

NISNet Winter School

Finse, May 2009

Audun Jøsang, UiO/UNIK – QUT

http://persons.unik.no/josang/

This presentation

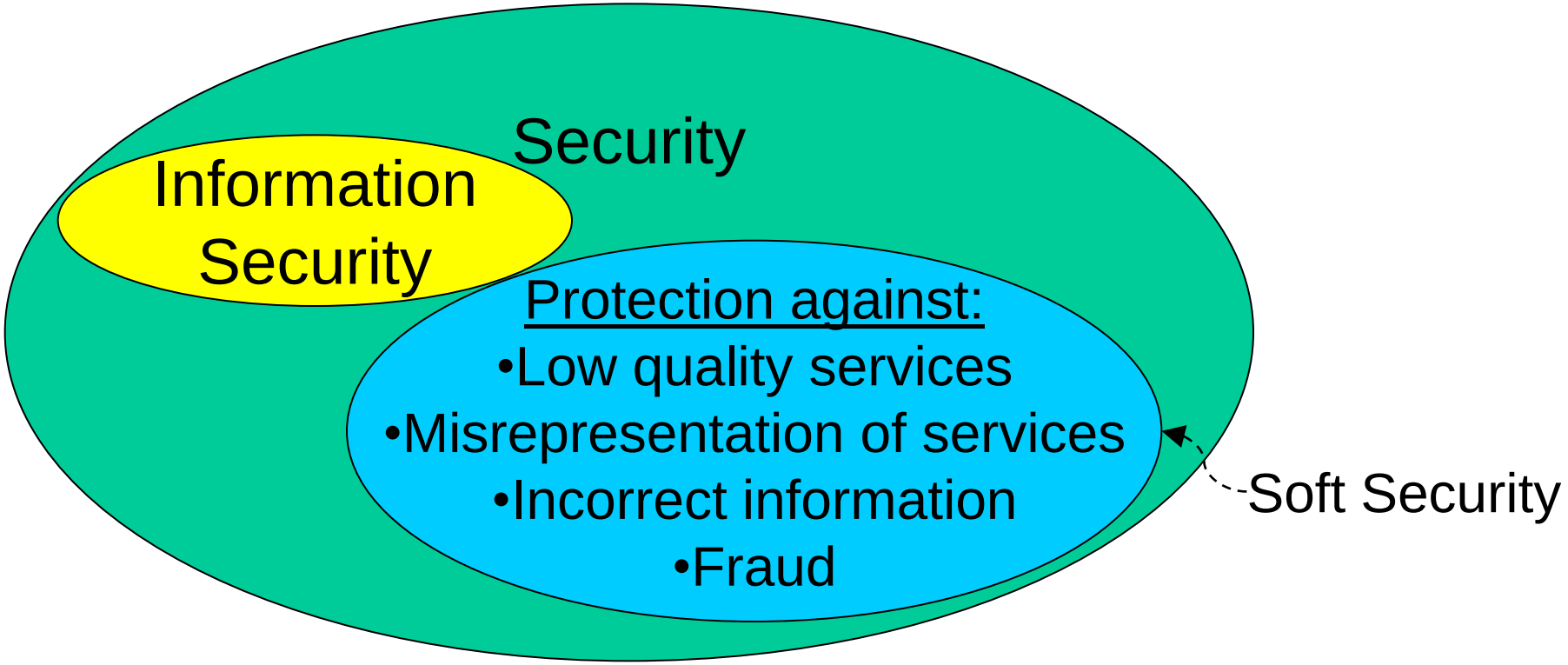
- Trust and Security
- Trust classes and trust semantics
- Principles for building trust and reputation systems
  - Network architectures
  - Computation engines
- Commercial and online systems
- Problems and proposed solutions
- Concluding remarks

Soft security and basic trust concepts

What is Security?

- General definition of security:
  - *Protection from danger*
  - Oxford English Online Dictionary: http://dictionary.oed.com/
- Traditional definition of information security:
  - *Preservation of confidentiality, integrity & availability of information*
  - ISO/IEC 27001:2005 Specification for an Information Security Management System
  - Assumes that the owner of information resources
    - defines a security policy (explicitly or implicitly)
    - implements measures to preserves CIA properties

Gap analysis of security and information security



Soft Security

- Impossible to define security policies for open communities
- Common ethical norms instead of security policy
  - Can be partly formal and partly dynamic/collaborative
- Definition:
  - ***Adherence to common (ethical) norms***
- Stimulates the quality of communities in terms of ethical behaviour and integrity of its members
- Enforced by collaborative mechanisms such as trust and reputation systems

Two definitions of trust

- Evaluation trust
  - The **subjective probability** by which an individual, *A*, expects that another individual, *B*, performs a given action on which its welfare depends. (Gambetta 1988)

- Decision trust
  - The **willingness to depend** on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible. (McKnight & Chervany 1996)

Would you trust this rope?

For what?

To climb down from the 3rd floor window of a house

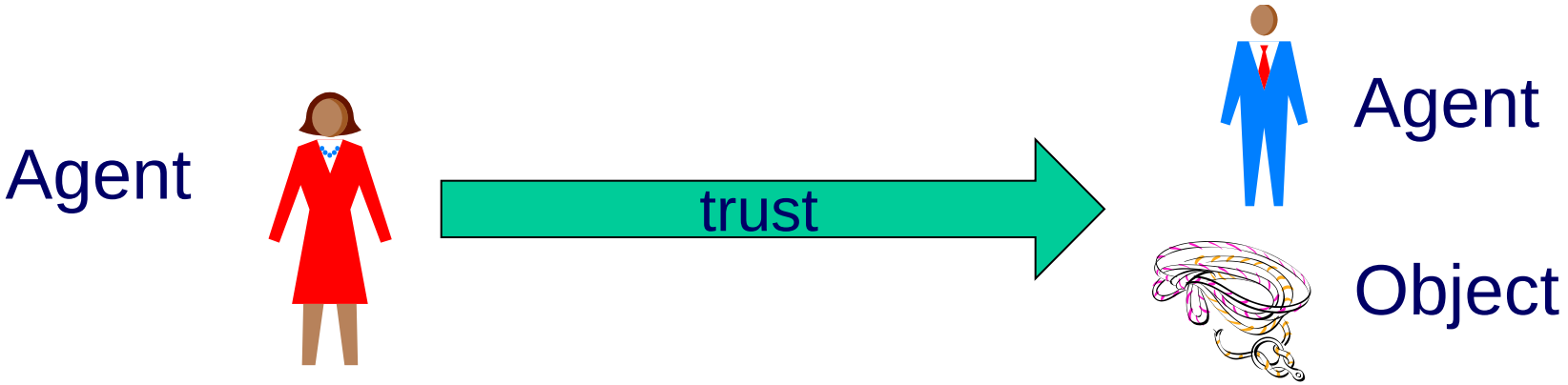
The rope looks very old

Fire drill: **No!**

Real fire: **Yes!**

UNIK UNIVERSITY GRADUATE CENTER

Trust is a relationship



- Trusting party
  - Also called
    - "relying party"
    - "trustor"
  - Is in a situation of
    - Dependence
- Trusted party
  - Also called
    - "trustee"
  - Is in a situation of
    - Power
    - Expectation to deliver

Two sides of trust management

Trusting party

Wants to **assess** and make **decisions** w.r.t. the dependability of the trusted party for a given transaction and context

Trusted party

Wants to **represent** and put in a **positive light** own competence, honesty, reliability and quality of service.

A definition of reputation

- Reputation is what is generally said or believed about a person's or thing's character or standing. (Concise Oxford Dictionary)
  - (Reputation of *B*)= Average[Reliability Trust in *B*]

Reputation and trust

REPUTATION

- Public info
- Common opinion
- Not necessarily objective

TRUST

- Both private and public info
- Private info carries more weight
- Subjective

- *"I trust you because of your good reputation"*
- *"I trust you despite your bad reputation"*

Extrinsic and intrinsic trust

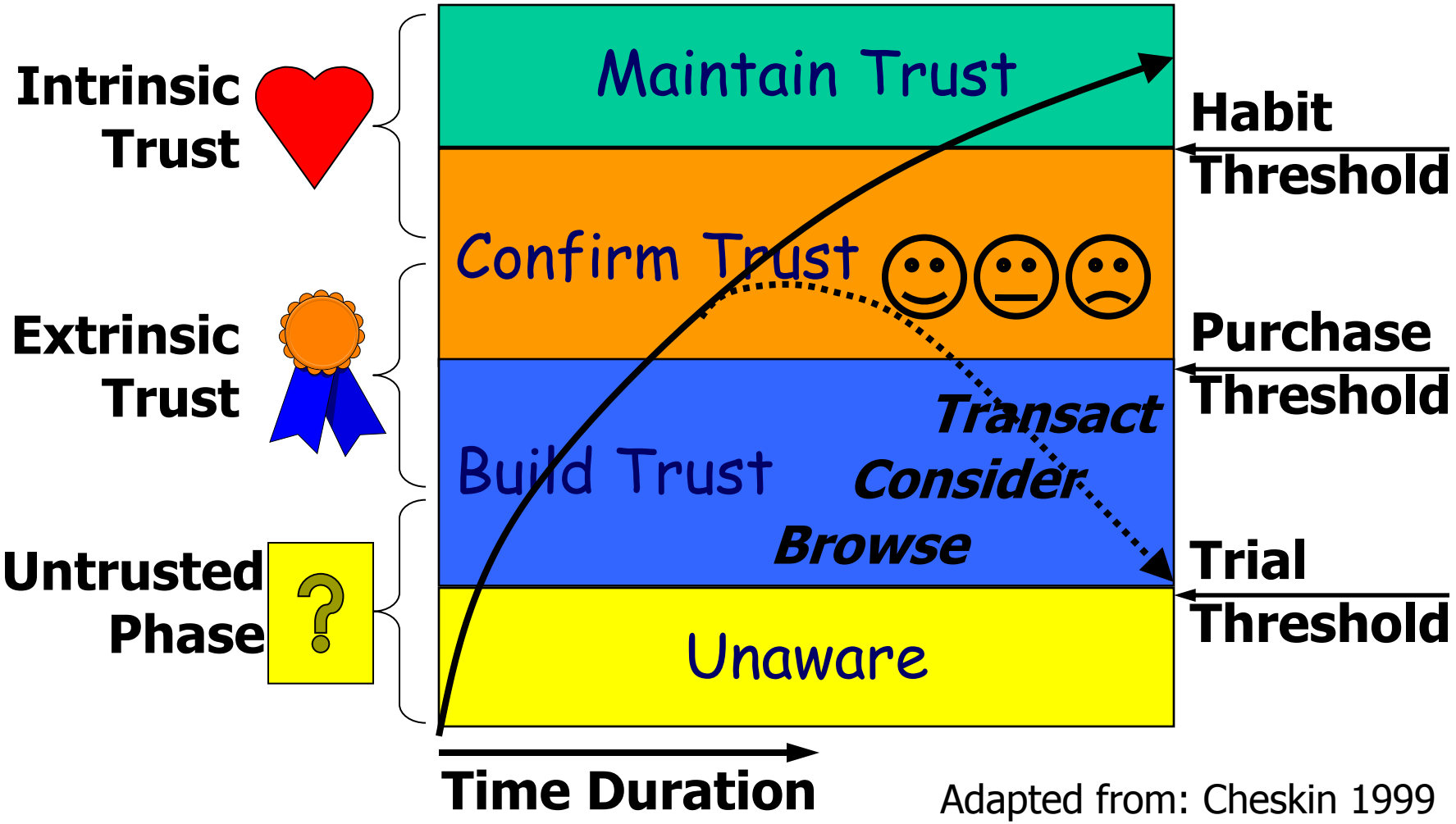
Extrinsic Factors

- Cognitive
- Observed
- Recommendation
- Reputation
- External evidence
- Easy to manufacture

Intrinsic Factors

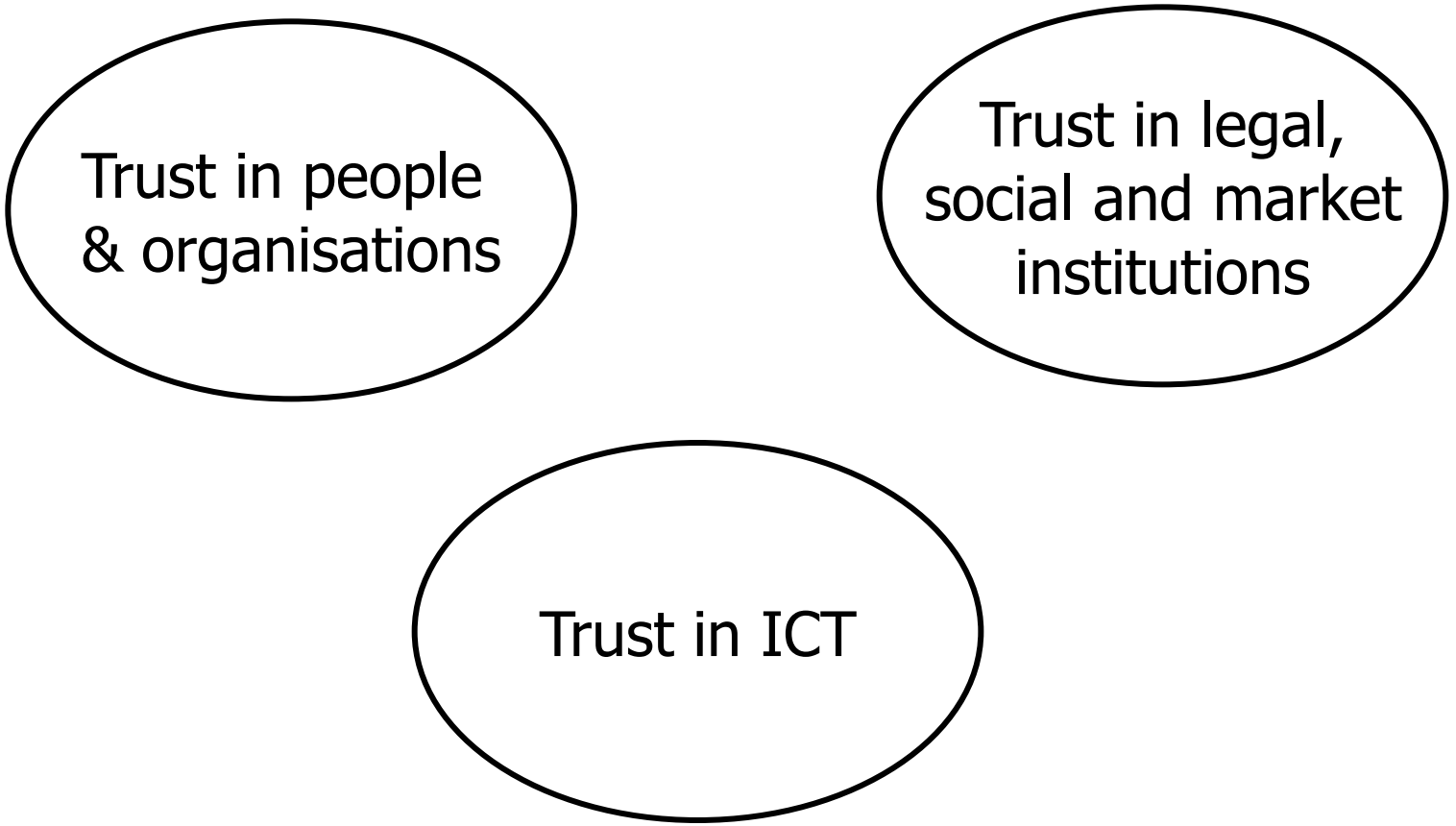
- Affective
- Experienced
- Intimate relationship
- Internalised pattern
- Take time to build
- Override extrinsic

A model for e-commerce trust

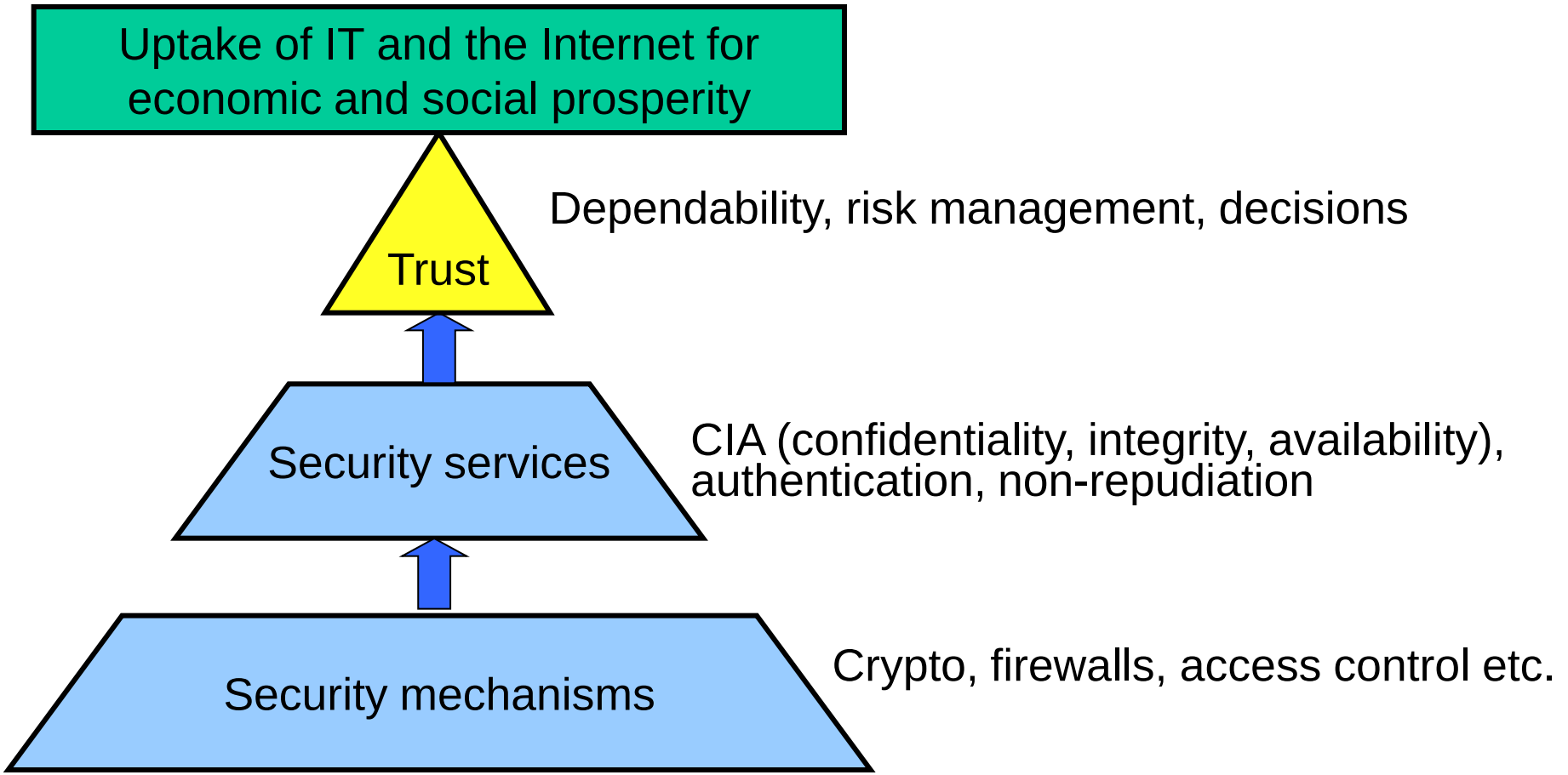


Adapted from: Cheskin 1999

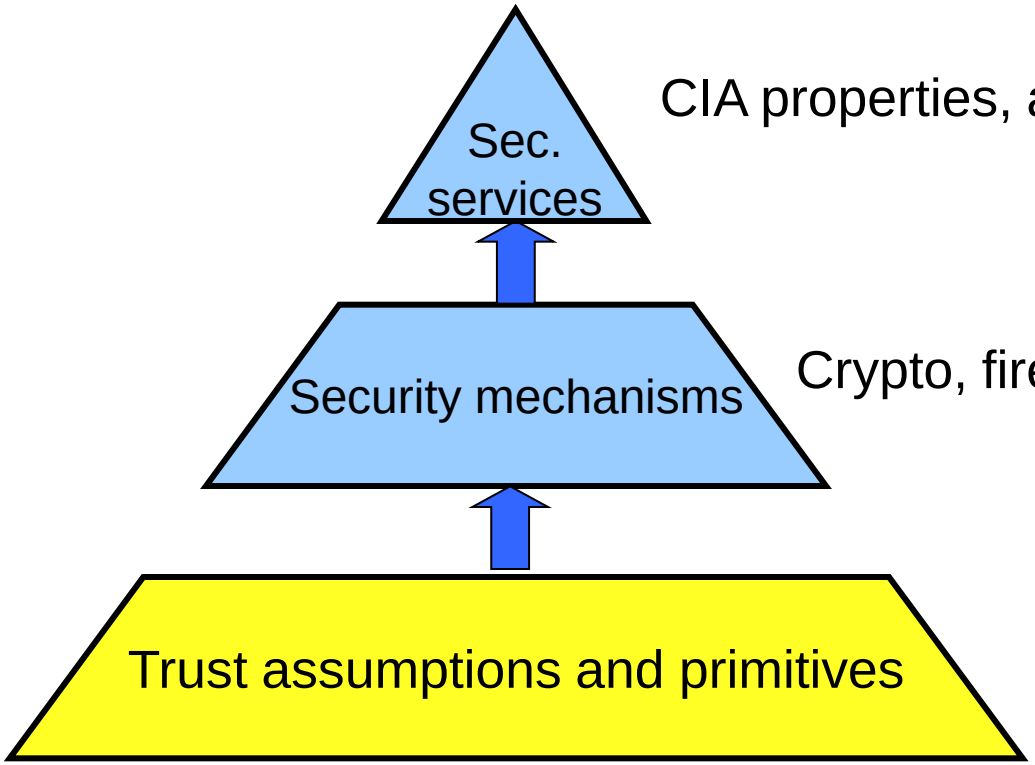
We trust what we depend on



Trust as an abstract security layer



Trust as assumptions and primitives



CIA properties, authentication, non-repudiation

Crypto, firewalls, access control etc.

- Algorithm strength
- Password confidentiality
- Platform security
- Certificates and credentials
- Tokens and attributes

Why is the term “trust” so popular?

- Metaphorical trust expressions
  - IT security people like metaphors:
    - E.g. firewall, honeypot, virus, Trojan horse, digital signature
  - Trust expressions serve as simple metaphors for complex security concepts, e,g. , …, ***trusted code, circle of trust***, …
- Trust has very positive connotations
  - Trust expressions are ideal as marketing slogans

Trust expressions can be difficult to intuitively understand

Trust Expressions in IT security

Trust management

Trustworthy computing

Trusted code

Trust bar

Trust anchor

Trust ecology

Trusted Computing Base

Trust system

Trusted system

Trusted computing

Trusted Platform Module

Computational trust

Trust negotiation

Trust model

Trust provider

Circle of trust

Trusted Third Party

Trust metric

UNIK
UNIVERSITY GRADUATE CENTER

Trust and access control

- Idea: “Who can I trust to access my resources?”
- Trusted user = authorized user
- Trusted code = code running as system
- Untrusted code = code running in a sandbox
- Access credentials can be exchanged and evaluated mechanically $\Rightarrow$ trust negotiation
- Access authorization can be delegated in a transitive fashion $\Rightarrow$ transitive trust

TC: Trusted Computing

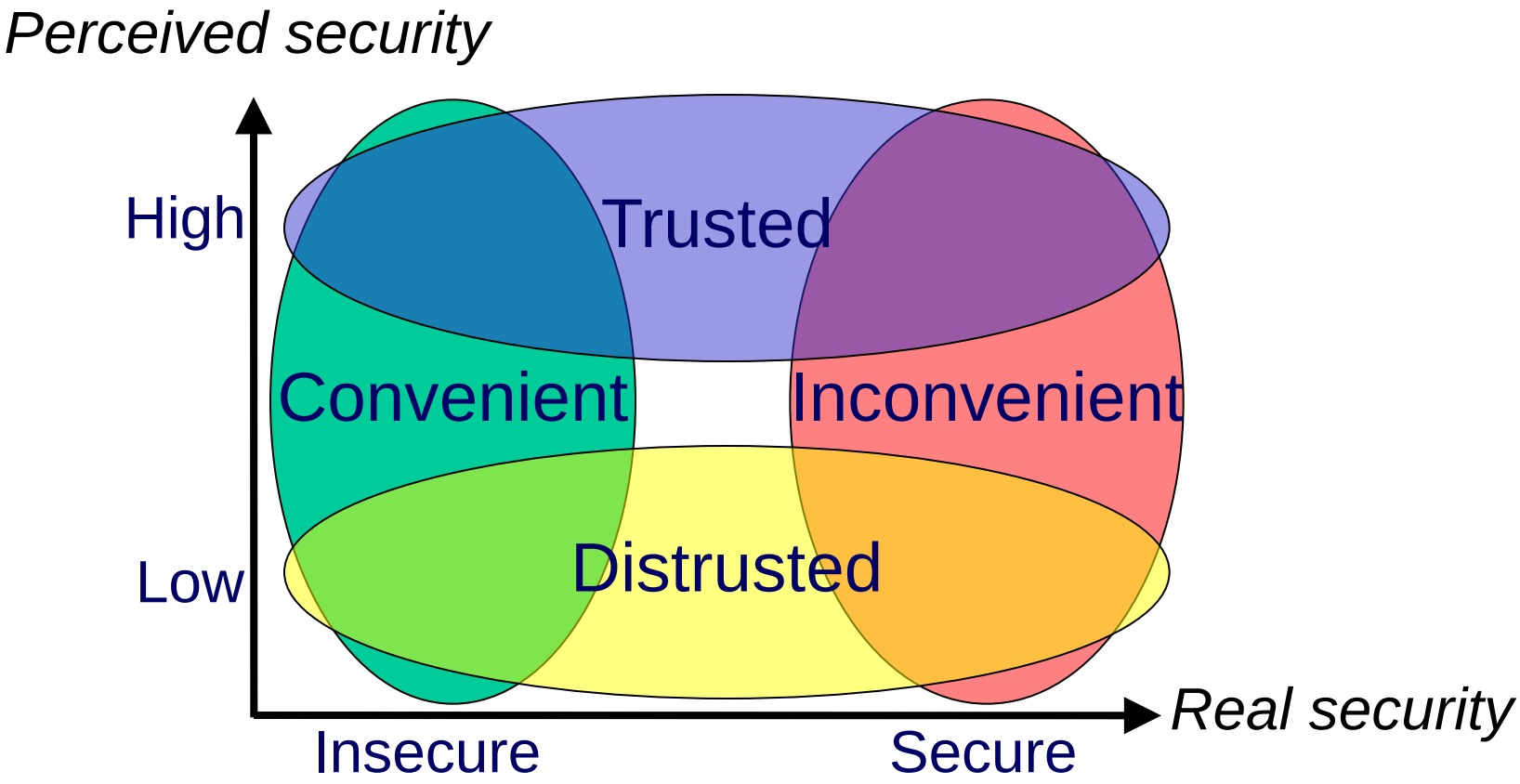
- Idea: Software can't be trusted, hardware can
  - Current paradigm: Security enforced by software
  - TC paradigm: Security enforced by hardware
- 1999: Trusted Computing Group (TCG)
  - Trusted Platform Module (TPM) specification
- 2001: Production of TPM chip
- 2002: Microsoft announces Palladium platform
  - Now: Next Generation Secure Computing Base (NGSCB)
- 2006: Limited trusted computing in Vista
  - Disk encryption based on TPM (trusted platform module)
- 2009: TPM in almost all PCs, not yet in mobiles

What trusted computing can do

- Can prevent
  - Installation and execution of unauthorised software
  - Tampering with installed software
  - Usage of stolen computers
- Can be used for Digital Rights Management (DRM)
  - Prevents playing unlicensed digital content

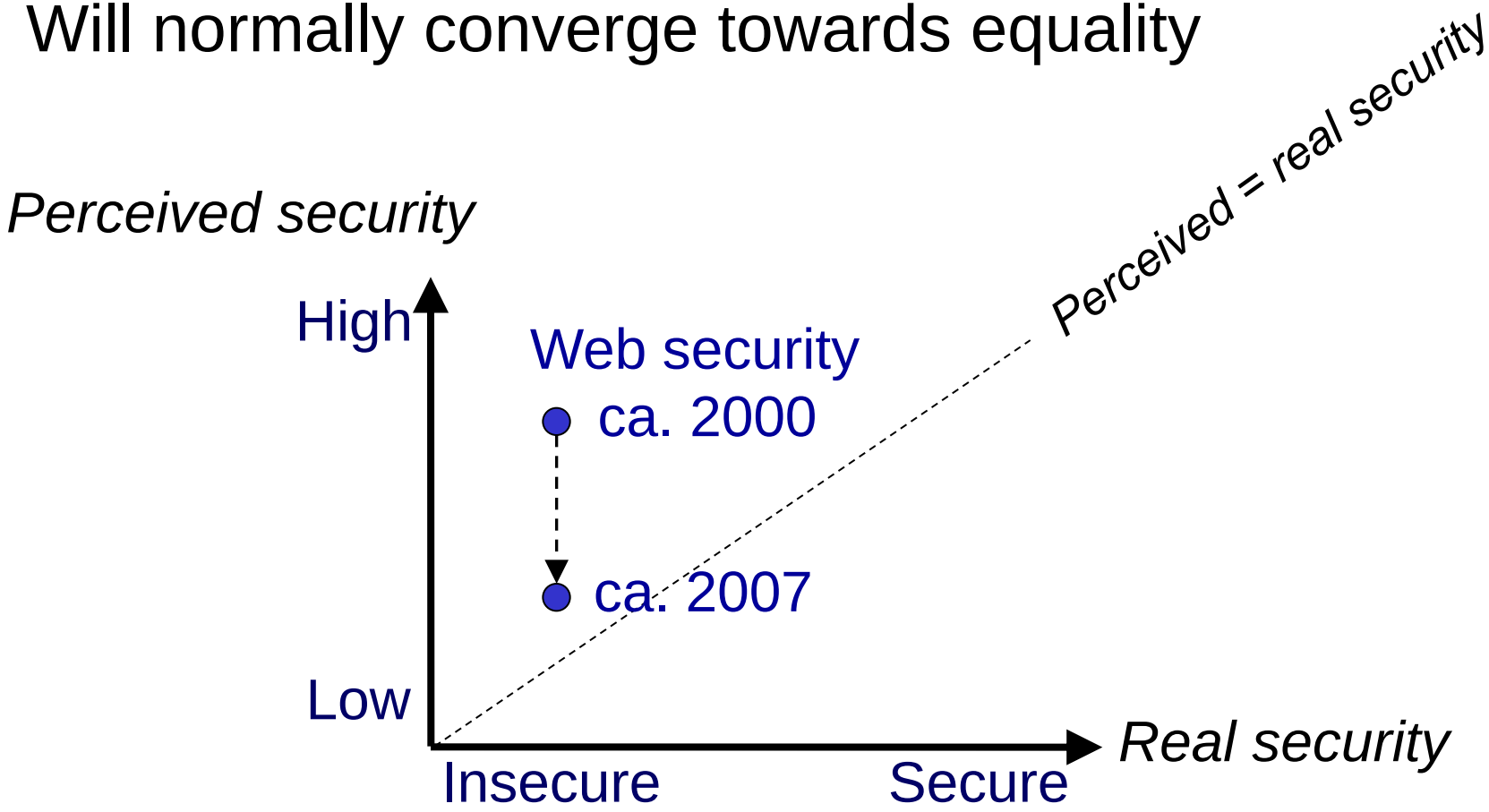
> If you want to do DRM on a PC, you need to treat the user as the enemy.
>
> Roger Needham
>
> Former director, Microsoft Research Europe

Perception and reality; The subjective perspective

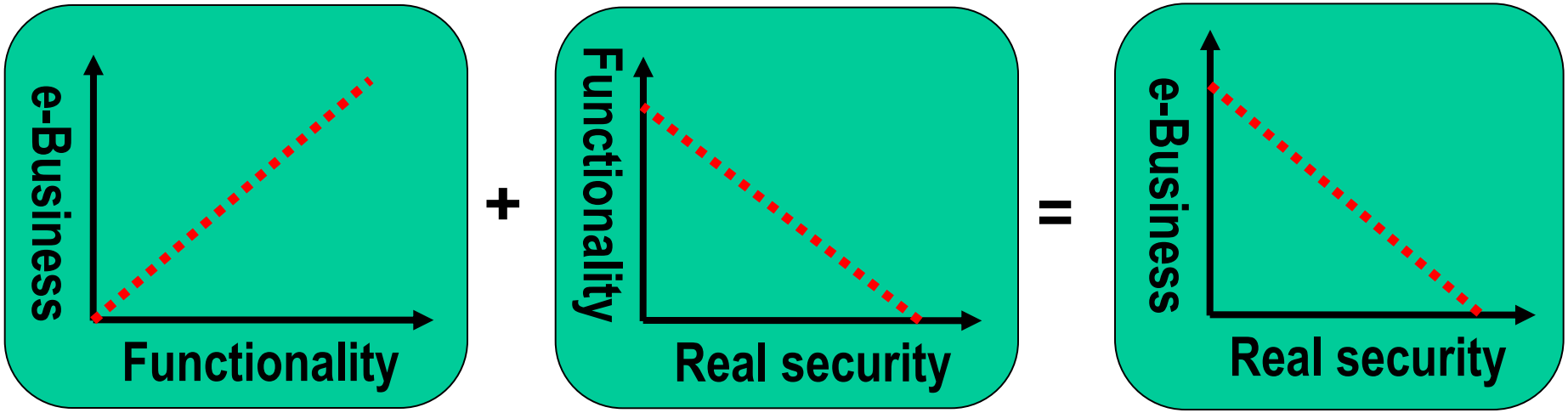


Real and perceived security

- Can temporarily be different
- Will normally converge towards equality

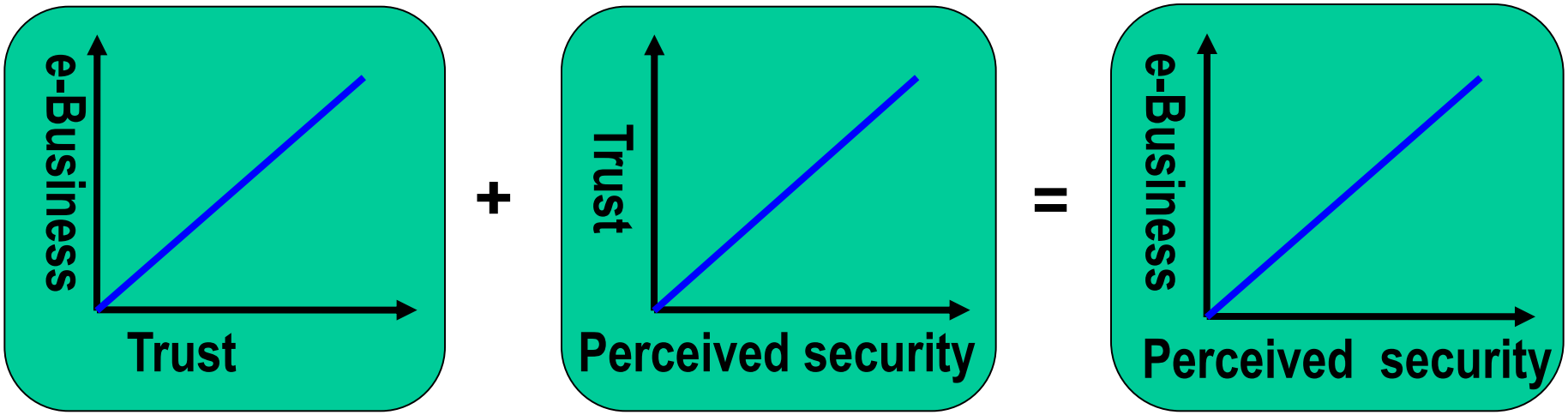


Real security is bad for e-business



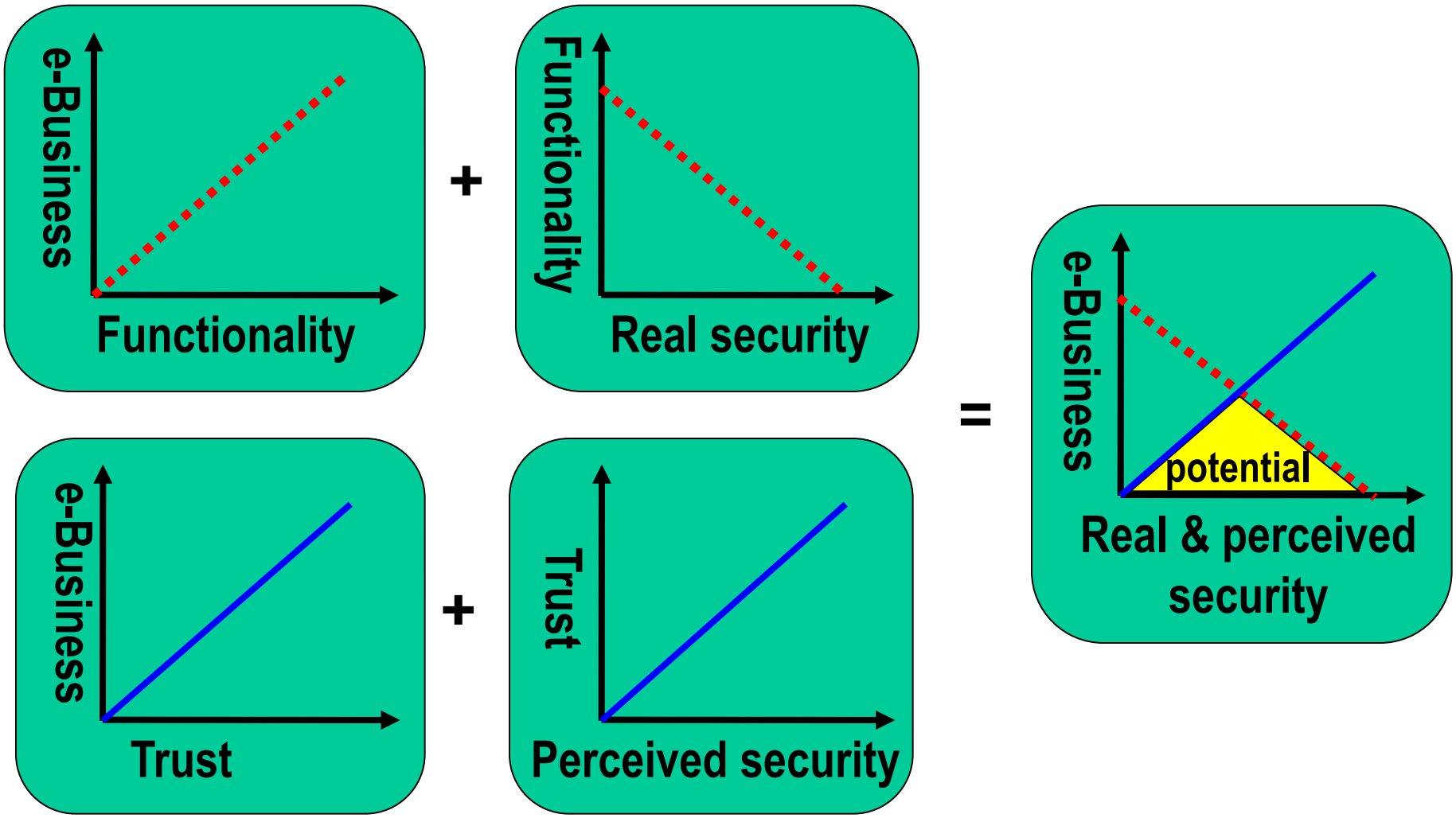
- e-business revolution not possible with real security
- Thank God the Internet isn't secure

Perceived security is good for e-business

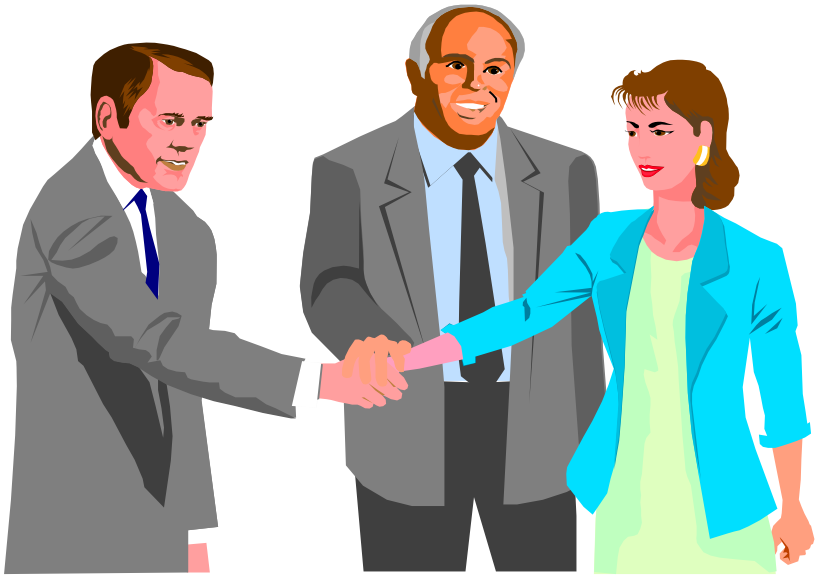


- e-business growth needs perceived security

e-Business growth potential



Trust classes and semantics



The trust scope: What we trust

Trust scope classes:

Service provision trust

–Relying party's trust in services and service providers.

Access trust

–Service provider's trust in users

Identity trust

–Belief that an entity's identity is as claimed

Delegation trust

–Trust in a agent to make trust decisions on behalf of the relying party

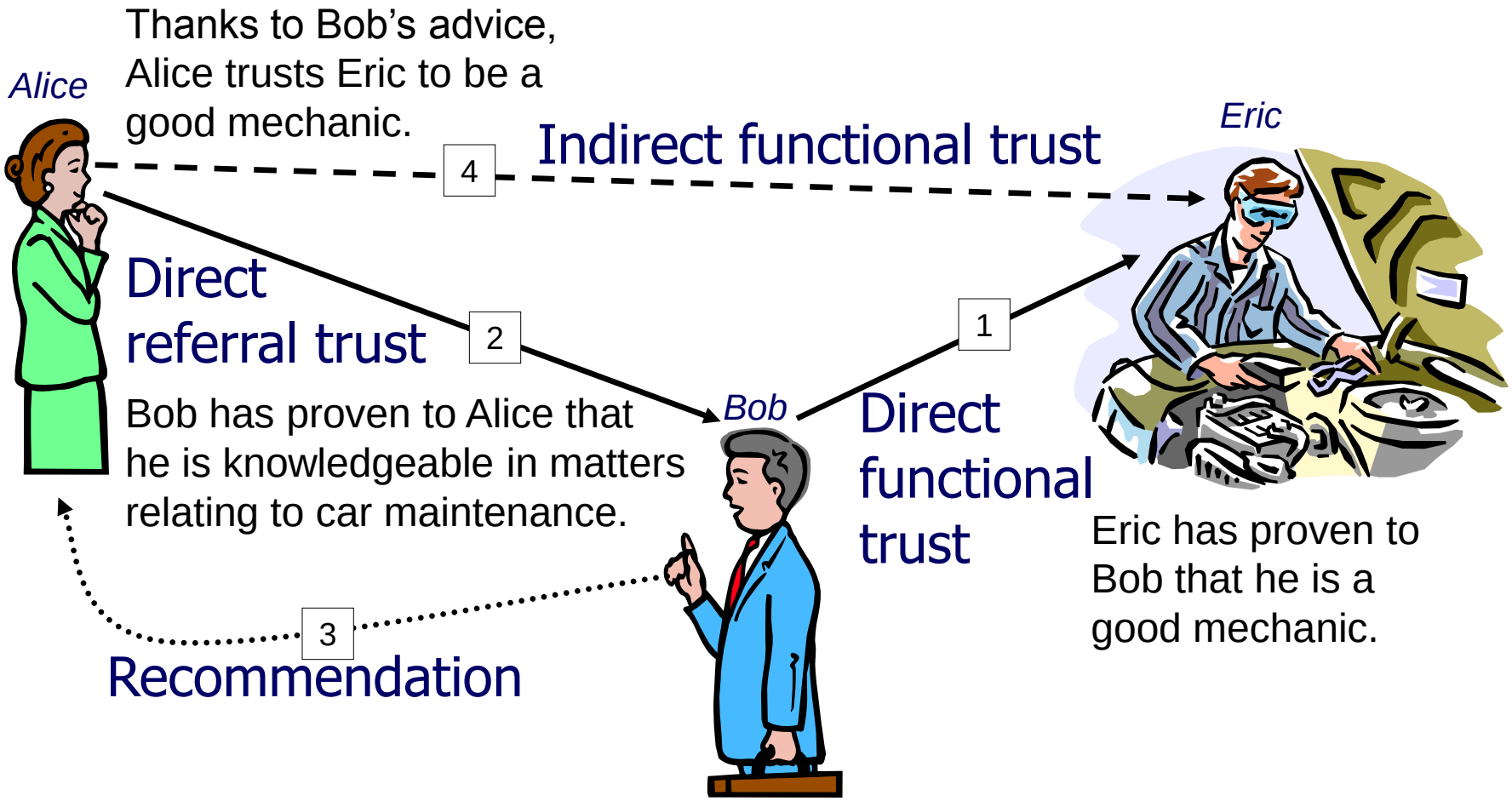
Context trust

–Belief that the necessary systems and institutions are in place in order to support a transaction that involves risk

Detailed trust semantics

- **Trust scope:** The combined set of functions that the relying party depends on and trusts
- **Functional trust:** The trusted party actually performs the functions of the trust scope
- **Referral trust:** The trusted party recommends a party (who recommends a party) that can perform the functions of the trust scope
- **Direct trust:** Derived from direct experience
- **Indirect trust:** Derived from recommendations

Trust transitivity

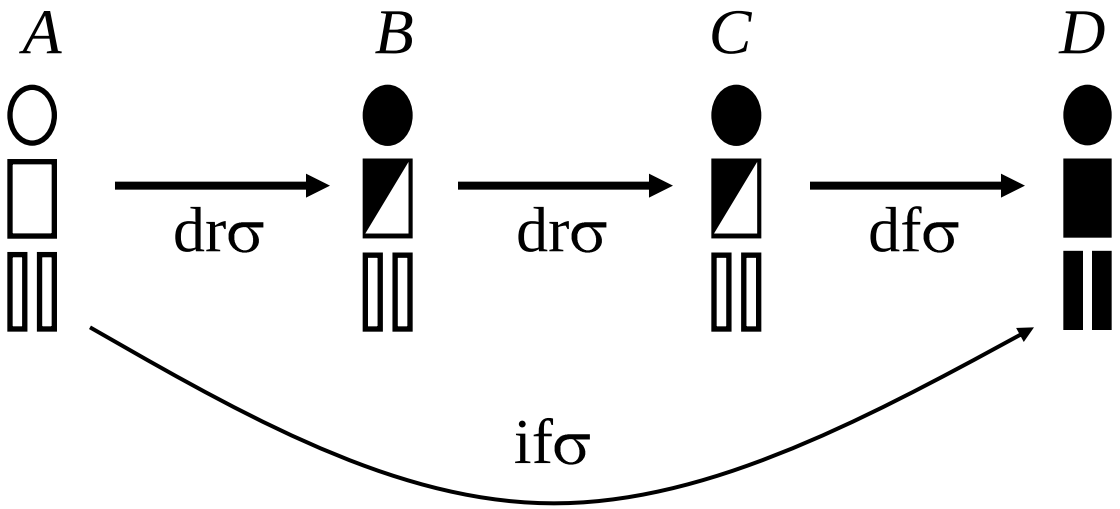


Additional trust dimensions

- Trust measure: μ
  - Binary (e.g. “Trusted”, “Not trusted”)
  - Discrete (strong-, weak-, trust or distrust)
  - Continuous (percentage, probability, belief)
- Time: τ
  - Time stamp when trust was assessed and expressed. Very important as trust generally weakens with temporal distance.

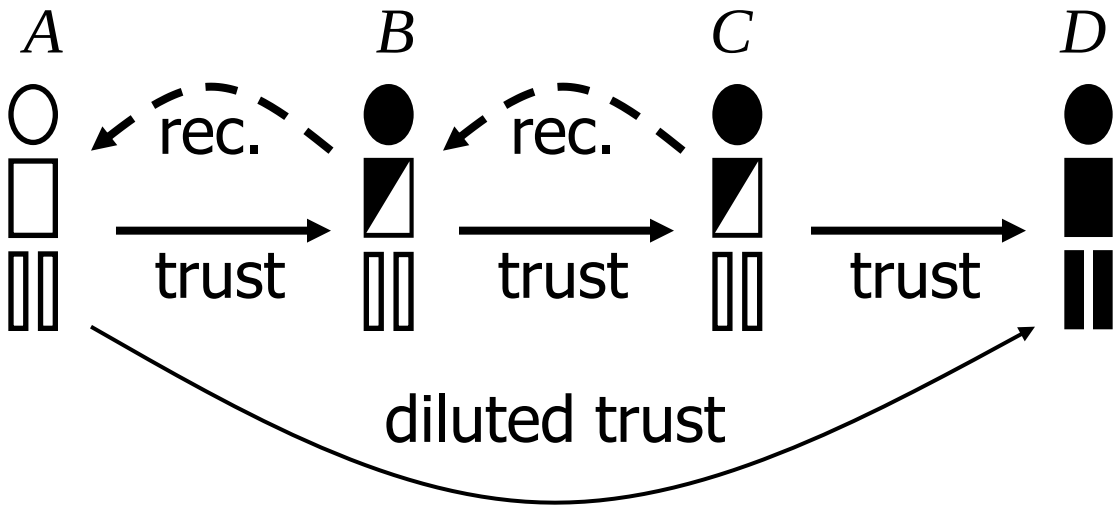
Valid transitive chains

- Every leg in the chain contains the same trust scope [σ]. (It doesn't make any sense otherwise!)
- The last trust link is **direct functional** trust [dfσ].
- All other trust links are **direct referral** trust [drσ].



Trust transitivity

Trust is diluted in a transitive chain.

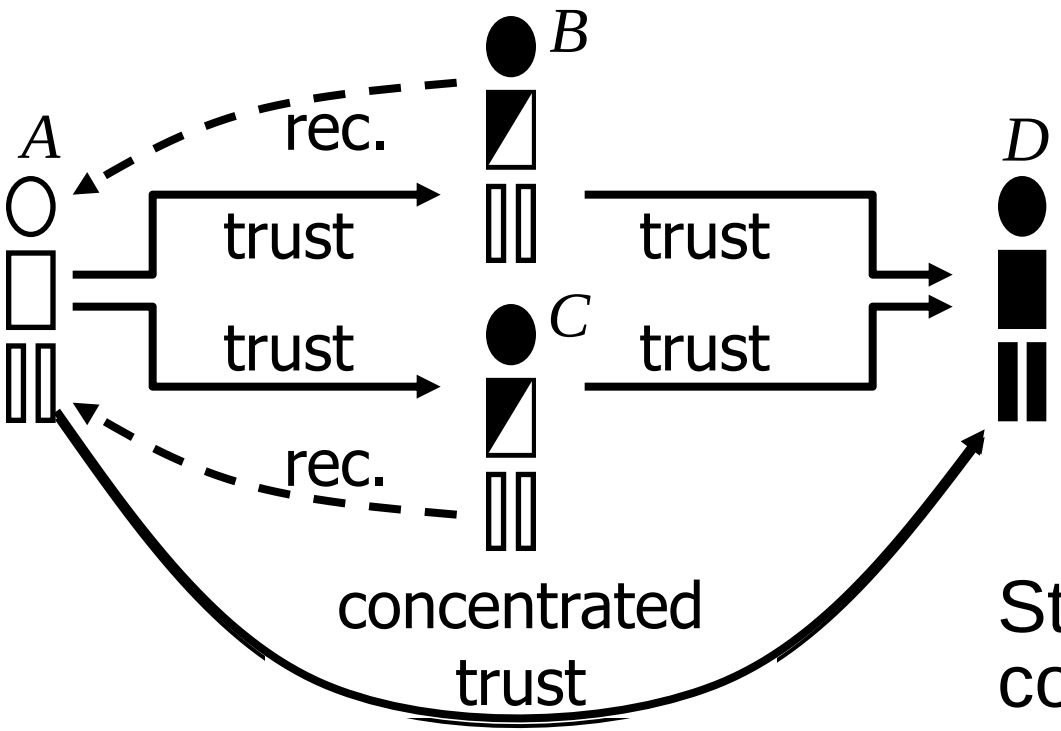


Computed with the transitivity operator of subjective logic

Graph notation: $[A, D] = [A, B] : [B, C] : [C, D]$

Explicit notation: $[A, D, \text{if}\sigma] = [A, B, \text{dr}\sigma] : [B, C, \text{dr}\sigma] : [C, D, \text{df}\sigma]$

Trust fusion

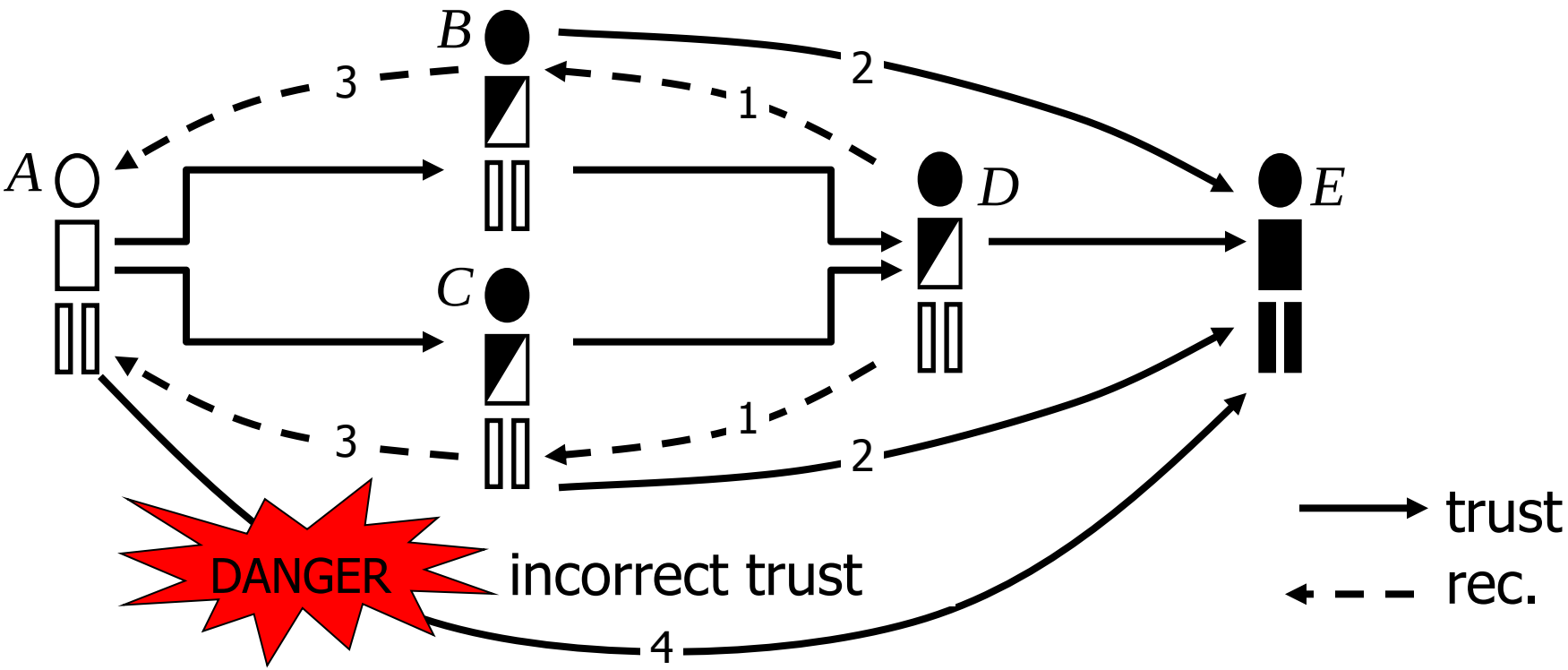


Strengthens trust confidence

Computed with the fusion operator of subjective logic

Graph notation: $[A, D] = ([A, B] : [B, D]) \quad ([A, C] : [C, D])$

Indirect referral trust

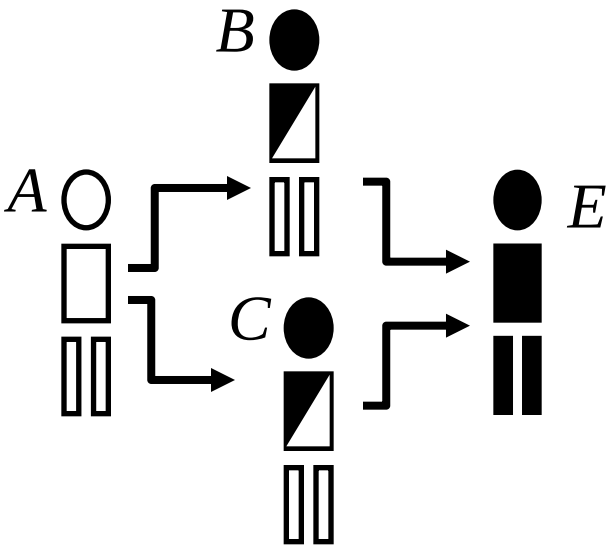


Perceived $[A, B] : [B, E]$ $[A, C] : [C, E]$ (OK)

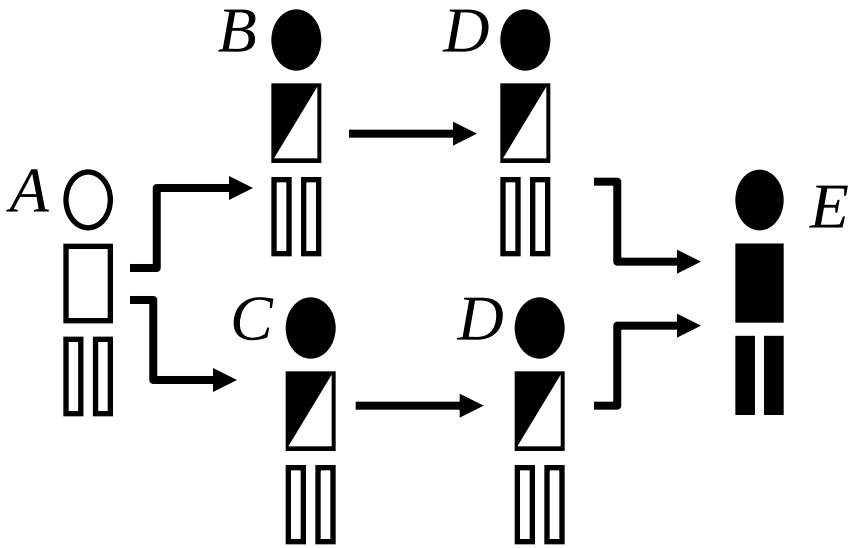
Reality: $[A, B] : [B, D] : [D, E]$ $[A, C] : [C, D] : [D, E]$ (not OK)

Hidden and perceived topologies

Perceived topology:



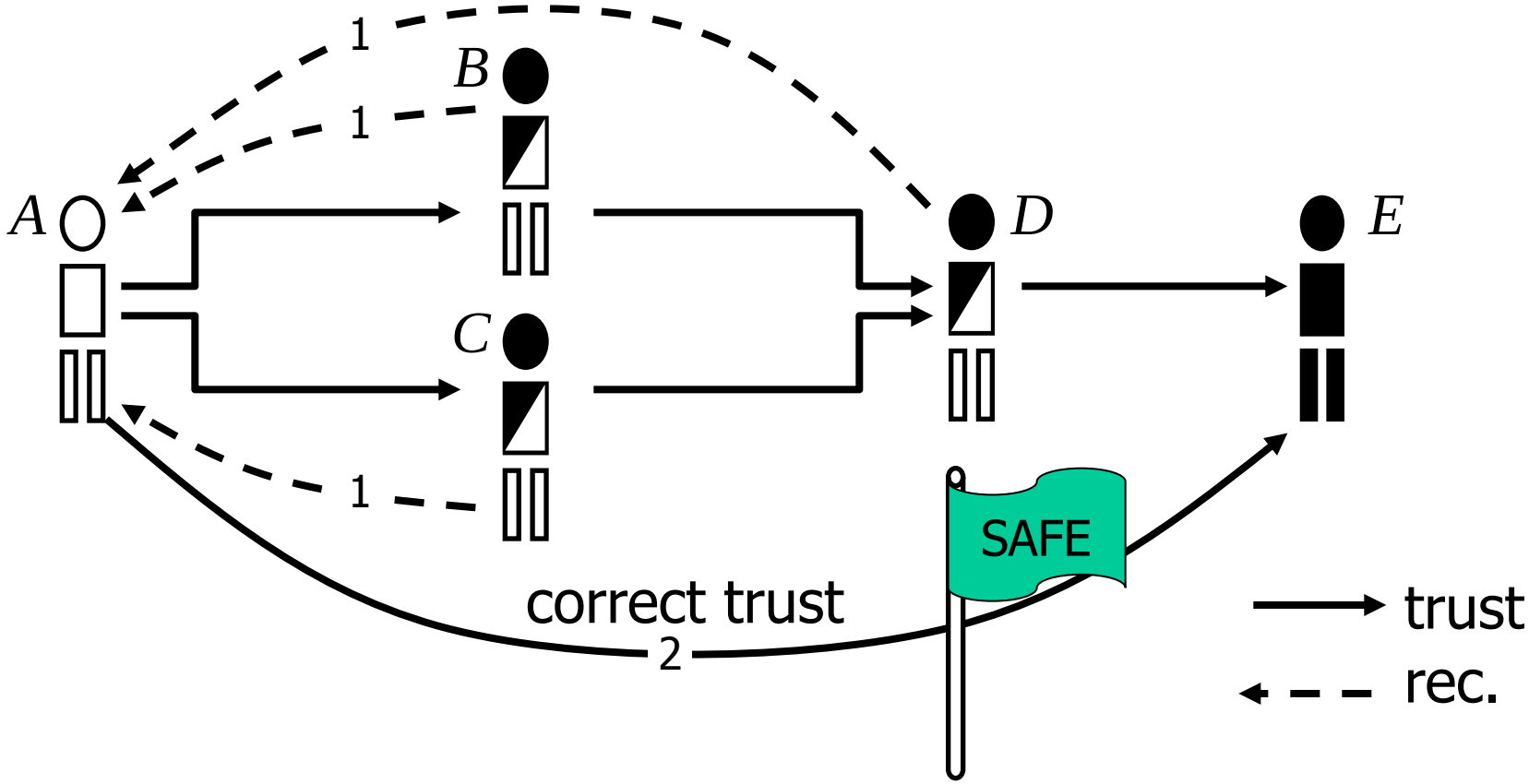
Hidden topology:



$$[A, B] : [B, E] \lozenge [A, C] : [C, E]$$
$$\neq [A, B] : [B, D] : \mathbf{[D, E]} \lozenge [A, C] : [C, D] : \mathbf{[D, E]}$$

(D, E) is taken into account twice

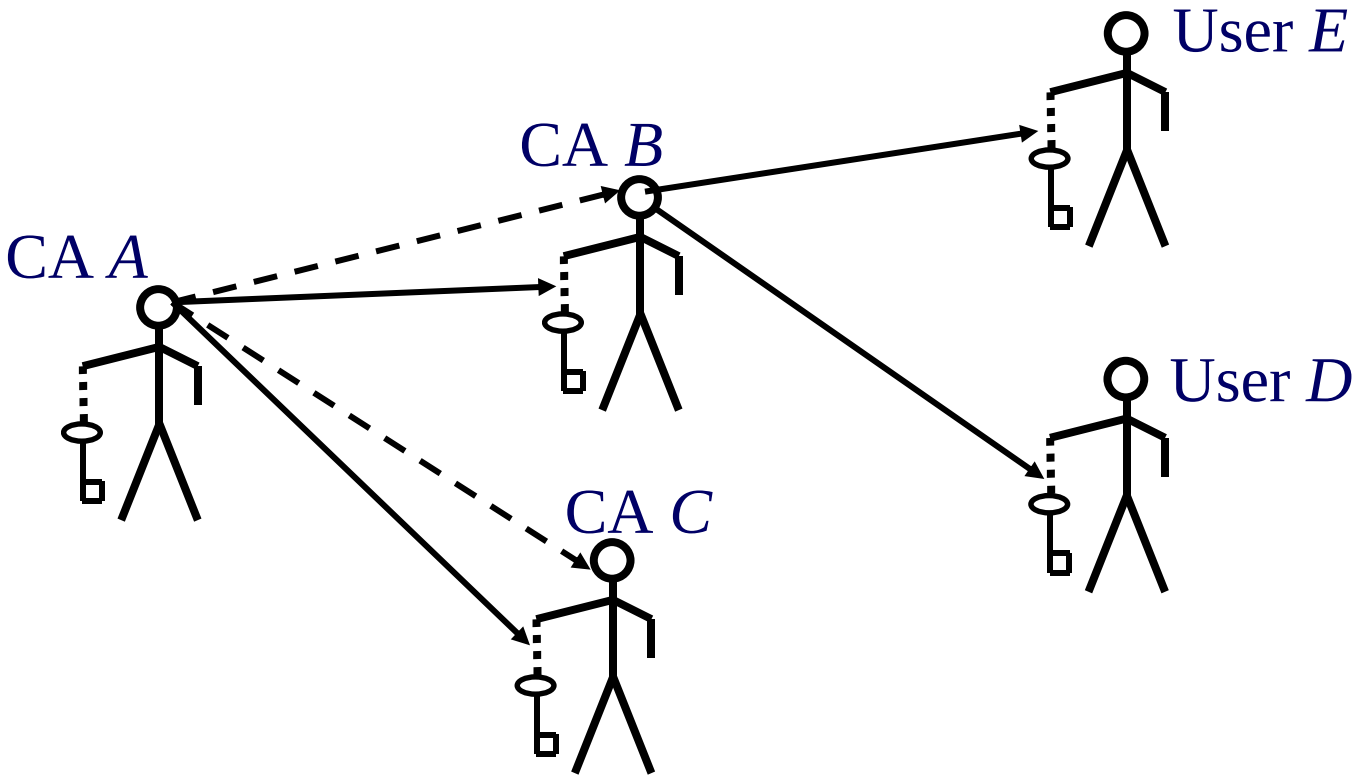
Correct indirect referral trust



Perceived and real topology (OK):

([A, B] : [B, D] [A, C] : [C, D]) : [D, E]

PKI and trust transitivity



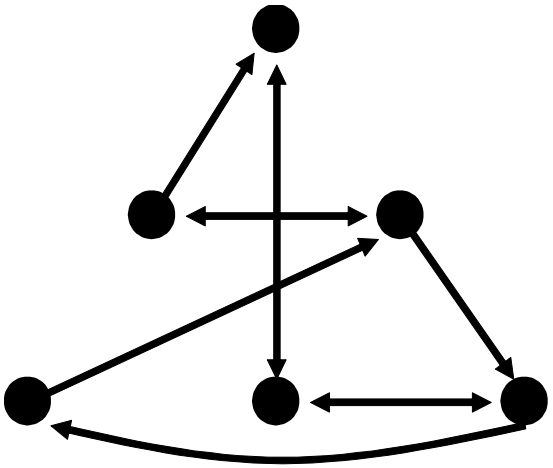
Trust in public keys (explicit through certificate chaining)

Trust in CA's (implicitly expressed through policies)

PKI with unstructured web of trust

- Uni- and/or bidirectional between arbitrary agents.
- No difference between CA and user.
- Not generally possible to authenticate all users.

- Example: PGP
  - Pretty Good Privacy



PGP web of trust

PGP trust model

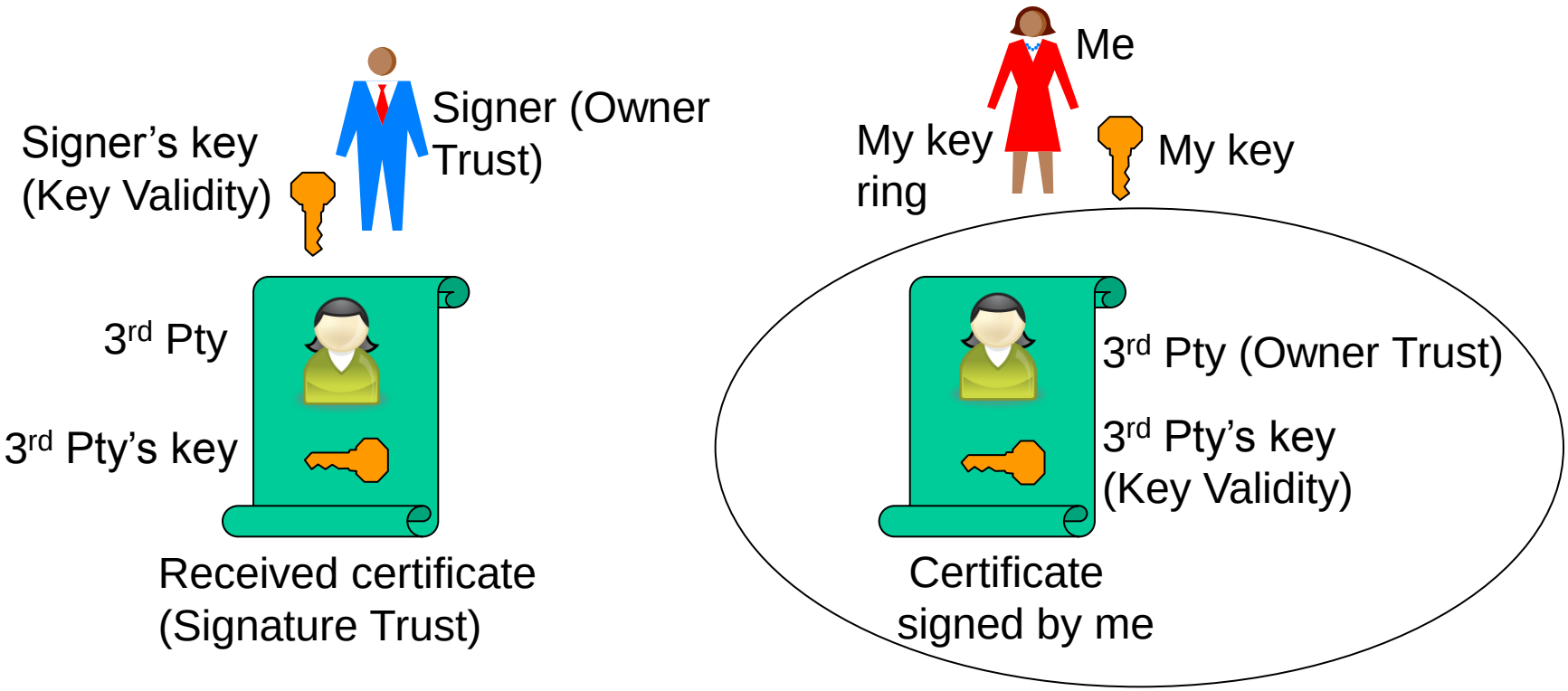
- ***Owner Trust:*** trust in the owner of a public key
- ***Signature Trust:*** trust in received certificates
- ***Key Validity:*** trust in a public key

Owner Trust:
Signature Trust:
- *always trusted*
- *usually trusted*
- *not trusted*
- *unknown trust*

Key Validity:
- *complete*
- *marginal*
- *undefined*

UNIK UNIVERSITY GRADUATE CENTER

PGP trust derivation overview



PGP trust derivation rules

1. Key Validity of received certificate signature key must be *complete*.
2. Signature trust := Owner Trust of signer
3. Owner trust of new key is manually set by Me
4. Key validity of new key is computed with Signature Trust values from one or several received certificates
5. By default PGP requires one *always trusted* or two *usually trusted* signatures in order to assign *complete* Key Validity
   - An insufficient number of *always trusted* or *usually trusted* signatures gives *marginal* Key Validity,
   - With no *usually trusted* signatures, Key Validity is se to *undefined*

Principles for building trust and reputation systems

Online v. brick and mortar world

	Availability and richness of trust evidence	**Efficiency of communication and processing**
Brick & mortar	Good	Poor
Online	Poor	Good

- Communication of trust information often restricted to local community in the real world
- The online world currently provides very little reliable trust evidence

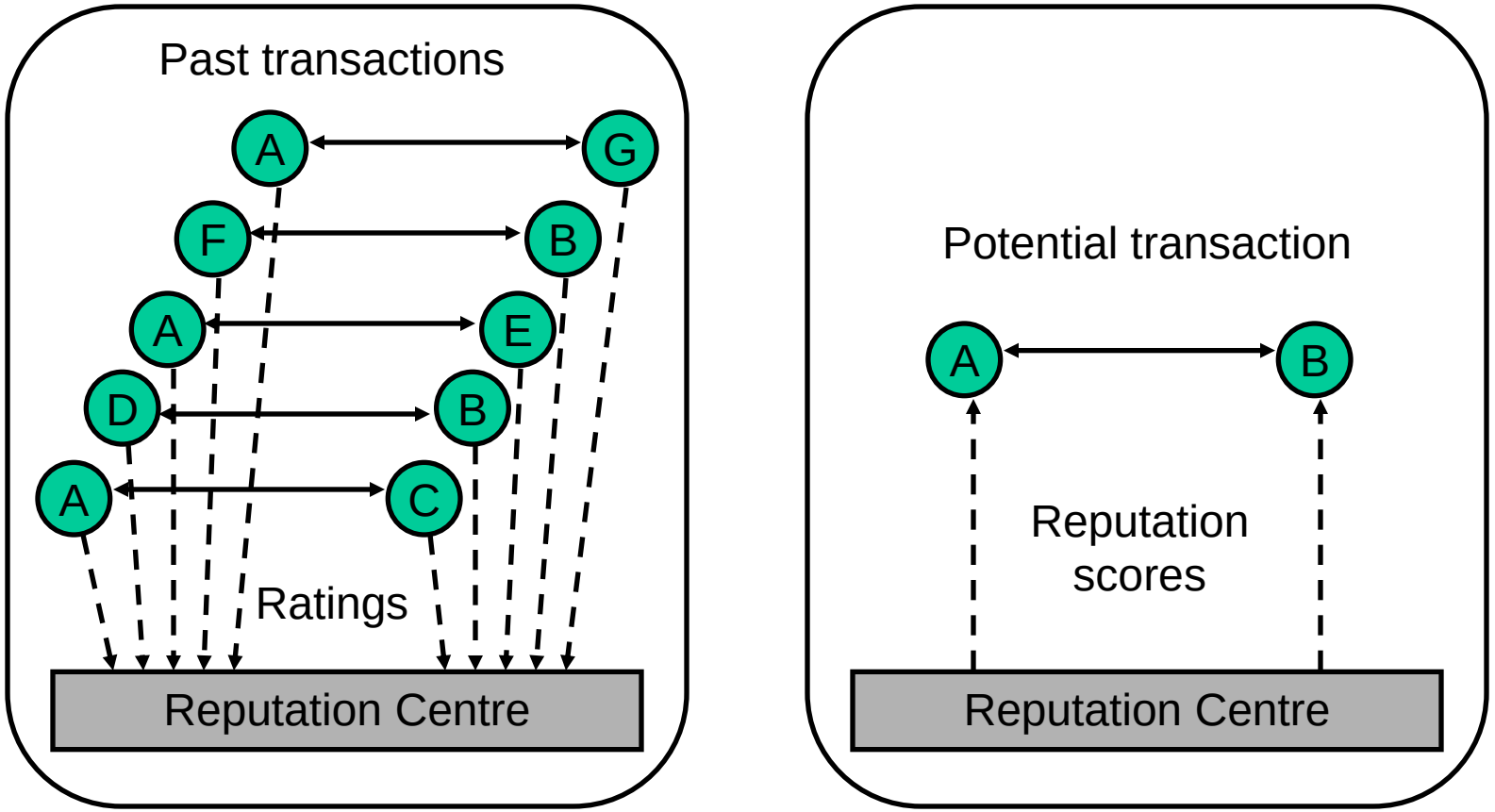
Basis for trust and rep. systems

- Focus on the trust evidence and on the methods for collecting this information
  - Find substitutes for traditional information used in physical world
  - Create new types of evidence
- Exploit the efficiency of IT and the Internet for
  - Collection of information
  - Processing
  - Dissemination

Trust/Reputation System Categories

	Private Scorers	Public Scores
Transitivity	Trust systems, e.g. Rummble.com	Public trust systems, e.g. PageRank
No transitivity	Private reputation systems, e.g. customer feedback analysis	Reputation systems, e.g. eBay.com

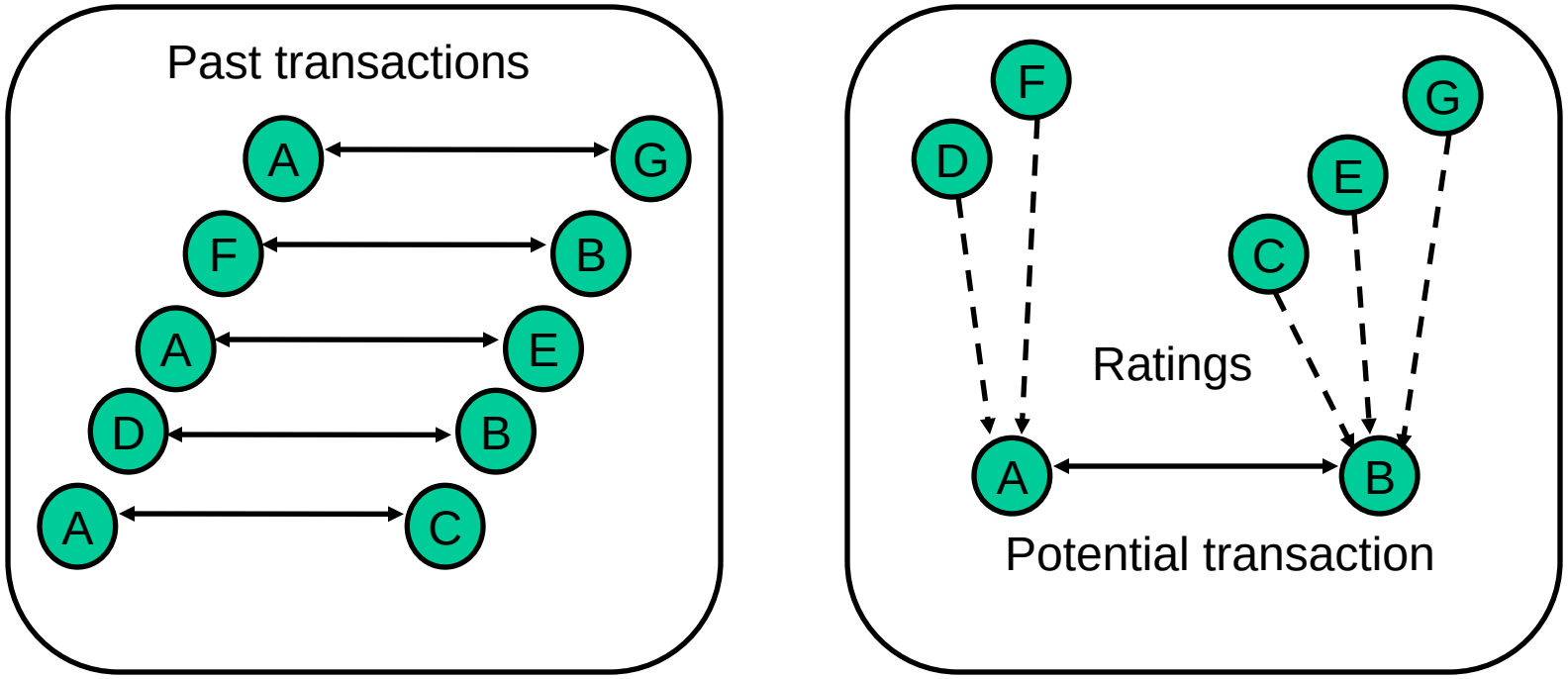
Centralised reputation system



a) Past

b) Present

Distributed reputation system



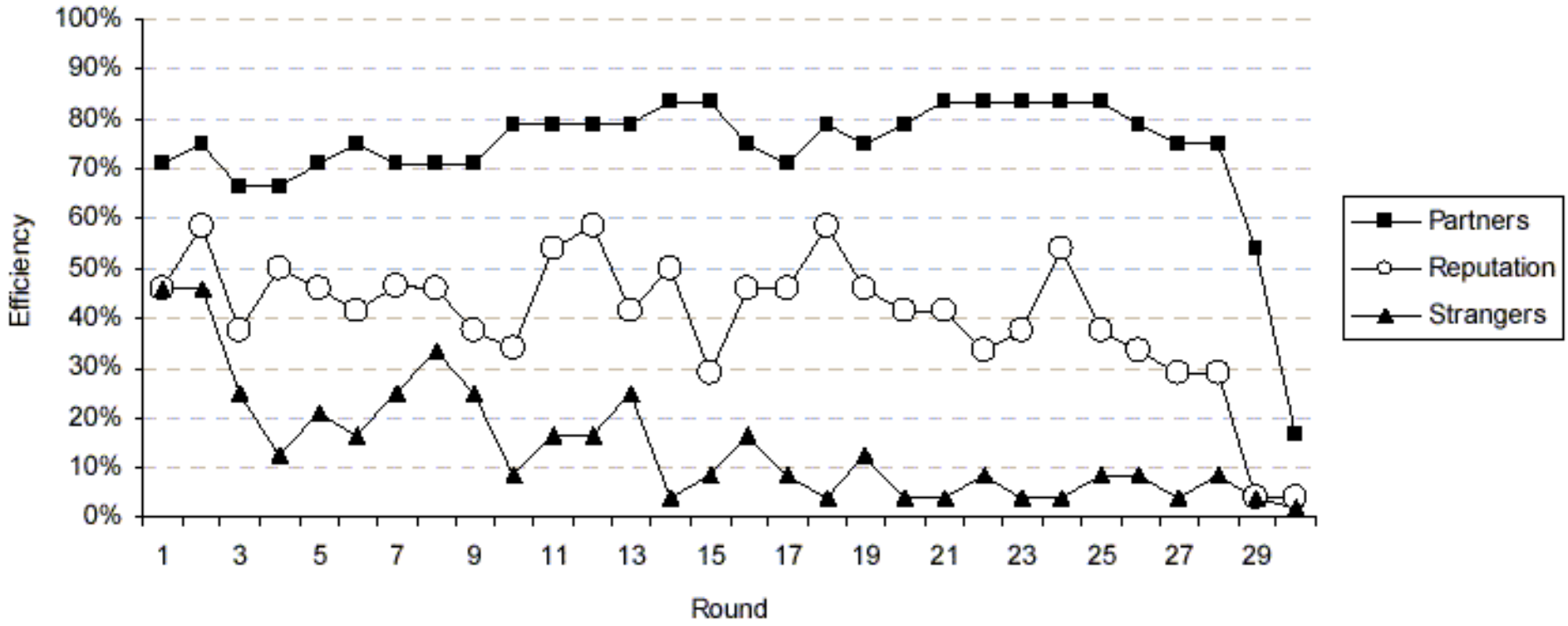
a) Past

b) Present

Applications

- e-Auctions
- P2P networks
- Software agent communities
- Contract negotiations
- Online markets: B2C, B2B, C2C
- Web service search and selection
- Information/intelligence gathering

Market Efficiency Experiment

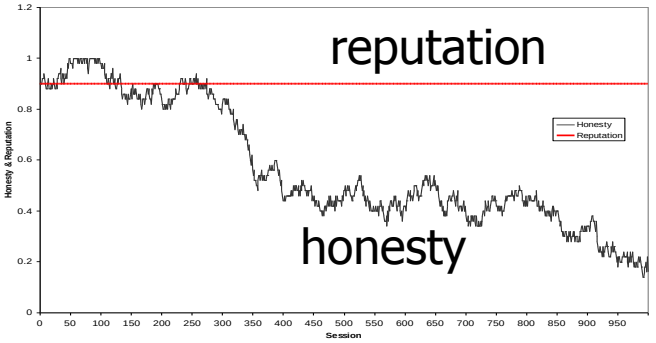


Source: Bolton,Katok,Ockenfels,2002

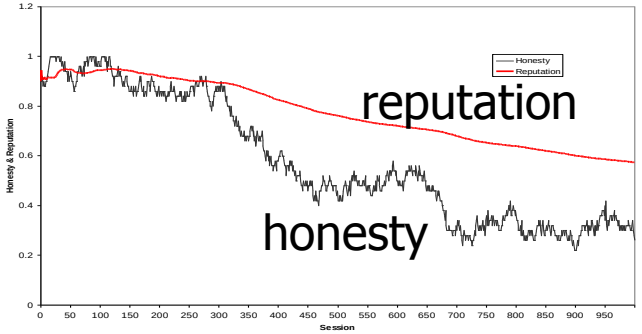
Simulation of the effect of reputation systems on e-markets

- Selling and buying software agents.
- Programmed to maximize profit by being dishonest.
- Reputation system enforces honesty

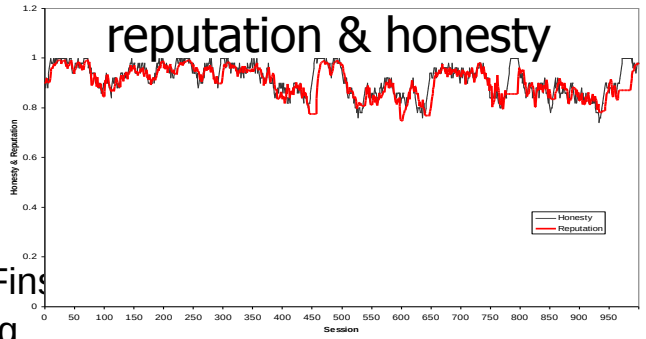
No ratings



Permanent ratings



Ageing ratings

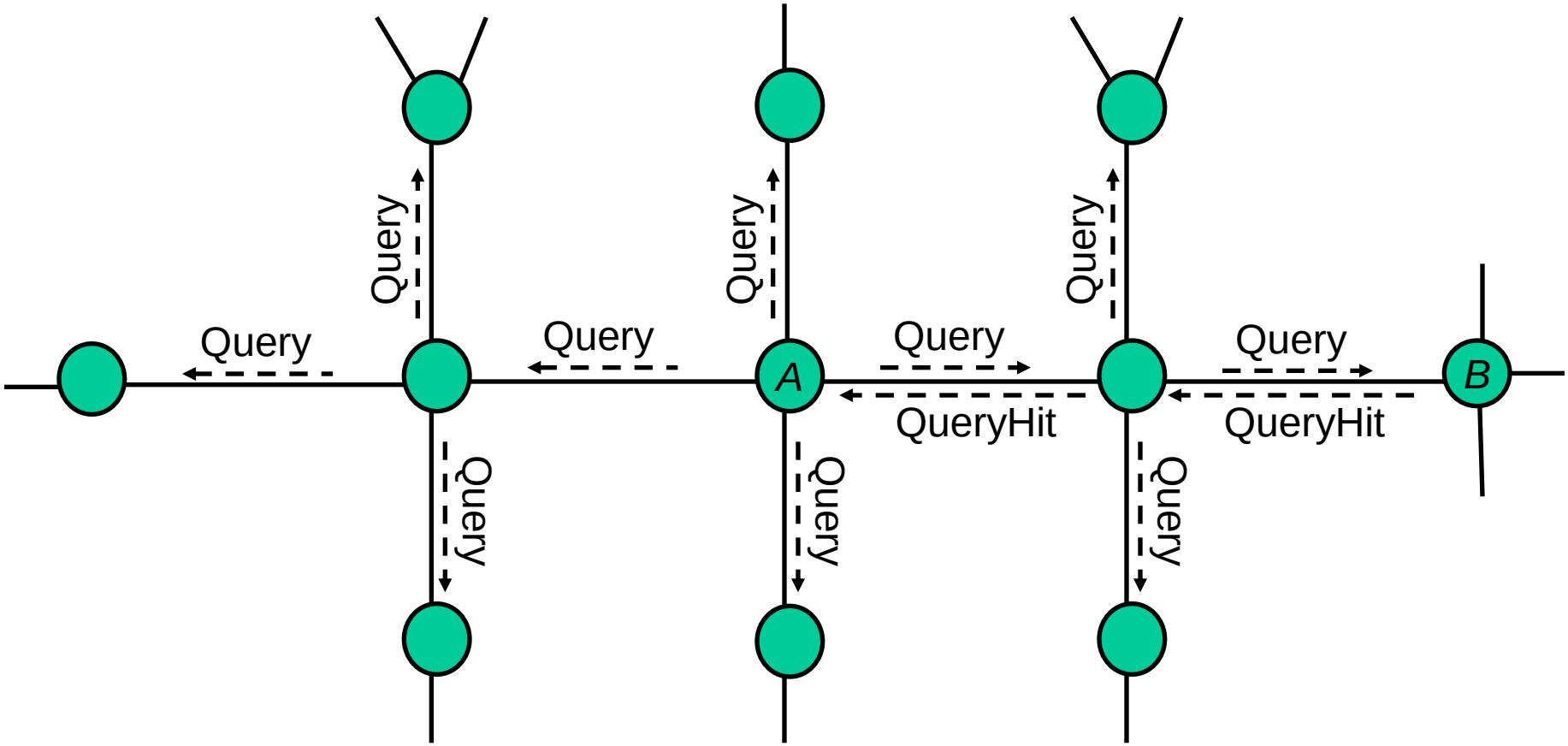


P2P networks

- P2P Networks: servent = server + client
- Search phase: discover resources
  - Centralised: e.g. Napster, with central directory
  - Pure distributed: Gnutella, Freenet
  - Semi-distributed: FastTrack, KaZaA, grokster, with distributed directory servers
- Download phase: get the resources
- Problems
  - Spreading malware
  - Free riding
  - Poisoning

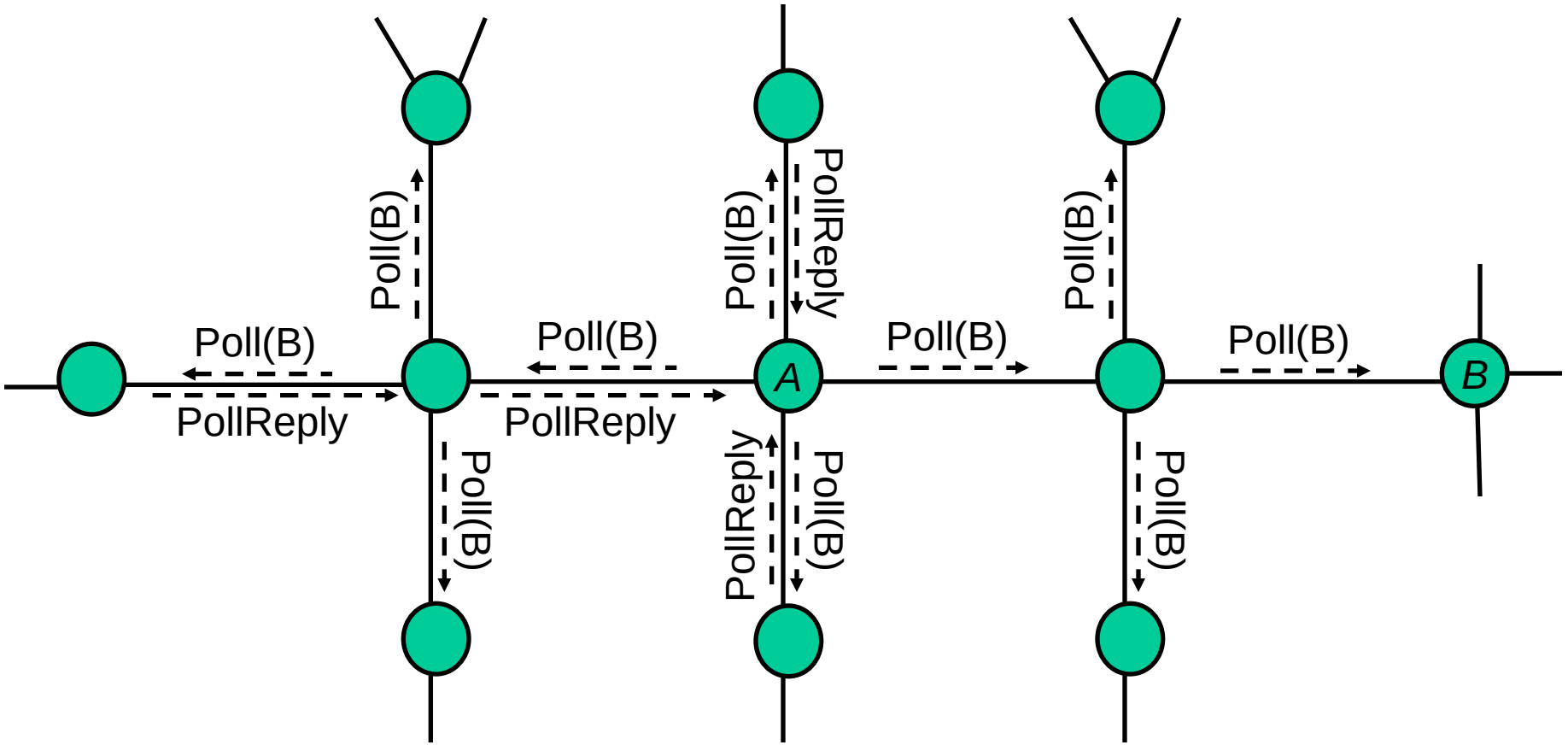
Gnutella example

- Pure distributed search phase



Reputation/trust system with Gnutella

- XRep proposed by Damiani *et al.*



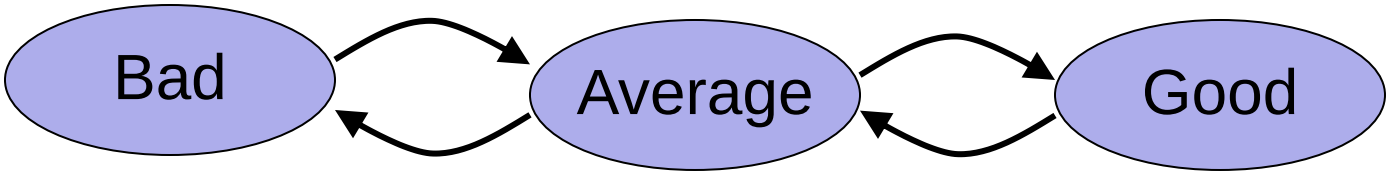
Trust and reputation computation engines

- Summation or average
- Hidden Markov
- Bayesian models
- Discrete models
- Belief models
- Fuzzy models
- Flow models

Summation and average

- Summation
  - Reputation score = Σ(positive) - Σ(negative)
  - E.g. eBay
- Average
  - Reputation score = Σ(ratings)/N(ratings)
  - E.g. Epinions
- Can be combined with sliding time windows
- Simple to understand
- Can give false impression of reputation

Hidden Markov Model



- True nature of future services unknown
- State of service/SP modelled as a Markov chain
- Statistically sound
- Requires parameters

Bayesian Reputation Systems

- Theoretically sound rating algorithm.
- Binomial and multinomial models.
- Rating possibilities:
  - any range,
  - combination,
  - discounting,
  - longevity,
  - weight ~ transaction value.

Computing binomial reputation over time with longevity factor

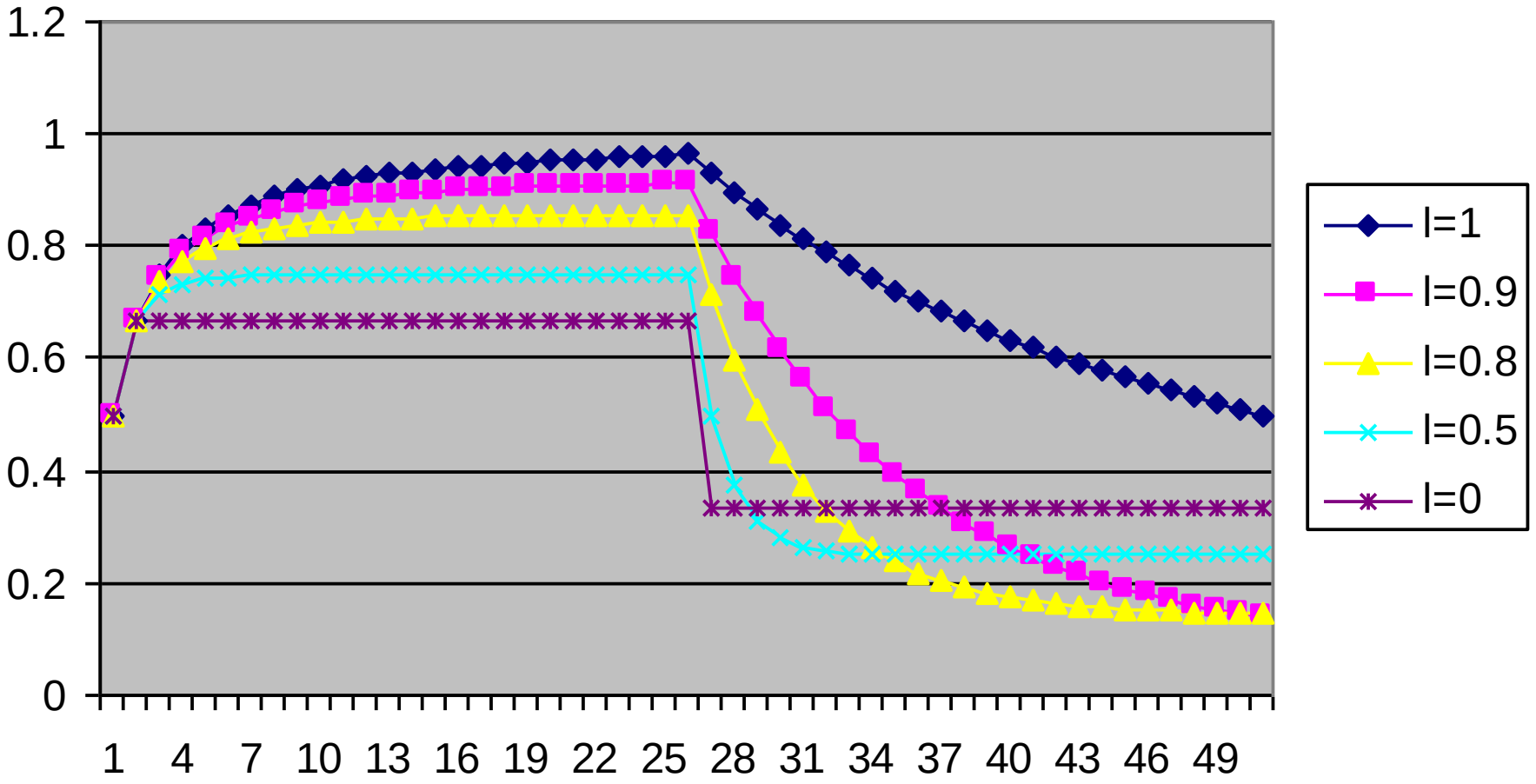
- R_i : accumulated positive evidence at time i
- S_i : accumulated negative evidence at time i
- r : positive evidence during 1 time period
- s : negative evidence during 1 time period
- λ : longevity factor in range [0,1]
- $R_{i+1} = \lambda \cdot R_i + r$: Recursive updating algorithm
- $S_{i+1} = \lambda \cdot S_i + s$: Recursive updating algorithm

- $$\text{Score}_i = \frac{r_{\text{base}} + R_i}{r_{\text{base}} + s_{\text{base}} + R_i + S_i}$$: Score at time period i
- Typically, $r_{\text{base}} = 1$, $s_{\text{base}} = 1$

Score evolution with different longevity

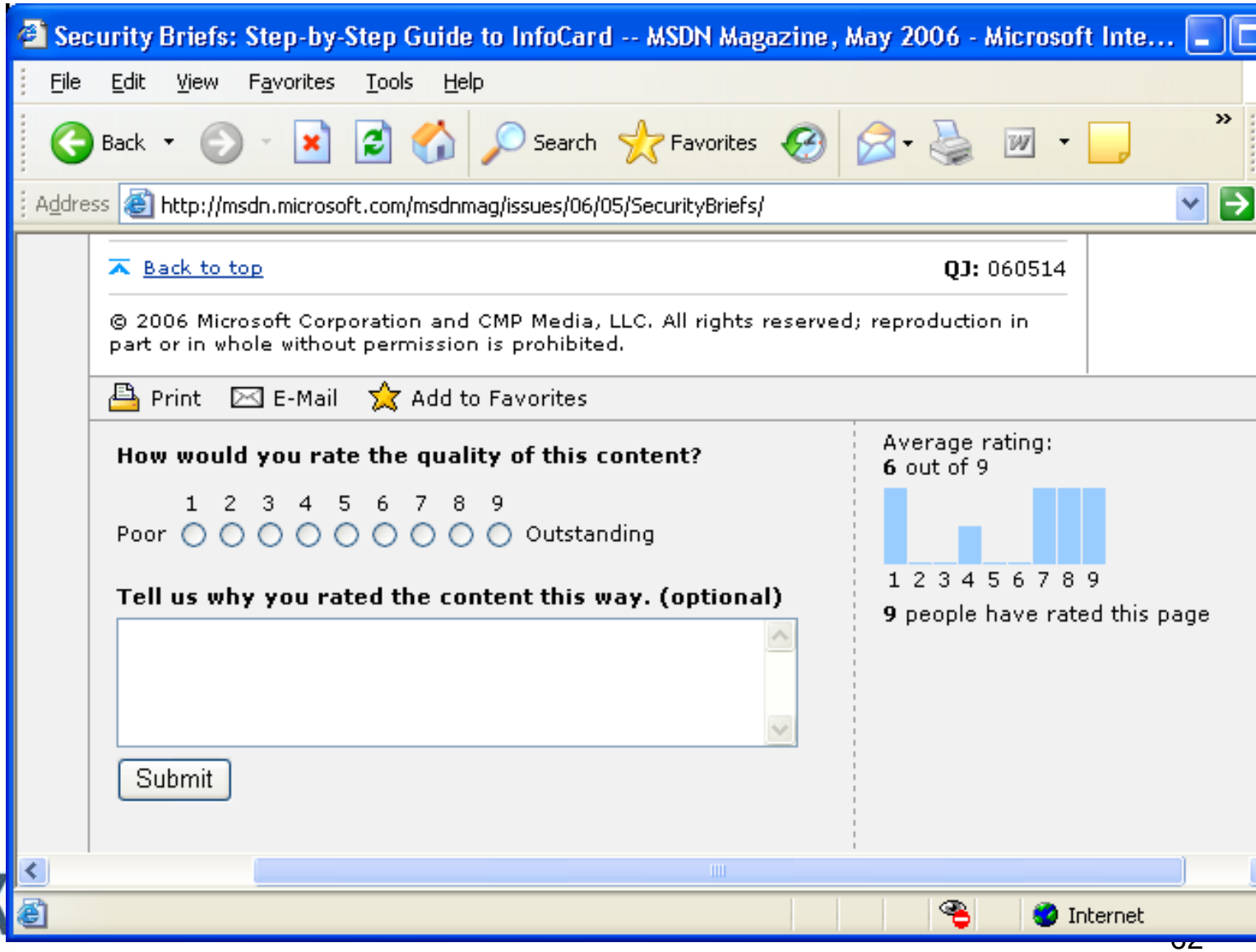
Period 1-25: Positive rating, $r = 1,\ s = 0$

Period 26-50: Negative rating, $r = 0,\ s = 1$



Multinomial reputation example

- Example from Microsoft
- Reflects polarised ratings



Multinomial reputation score

- The multinomial reputation score can be defined equal to the Dirichlet-PDF probability expectation

$$\text{Score}\,(L_j \mid \vec{r}, \vec{a}) = \frac{r(L_j) + C \cdot a(L_j)}{C + \sum_{j=1}^{l} r(L_j)}$$

Rep. score

$\vec{r}$: Multinomial evidence vector

$\vec{a}$: Multinomial base rate vector

$C = 2$

l : Number of rating levels

L_j : particular rating level

Initial reputation score

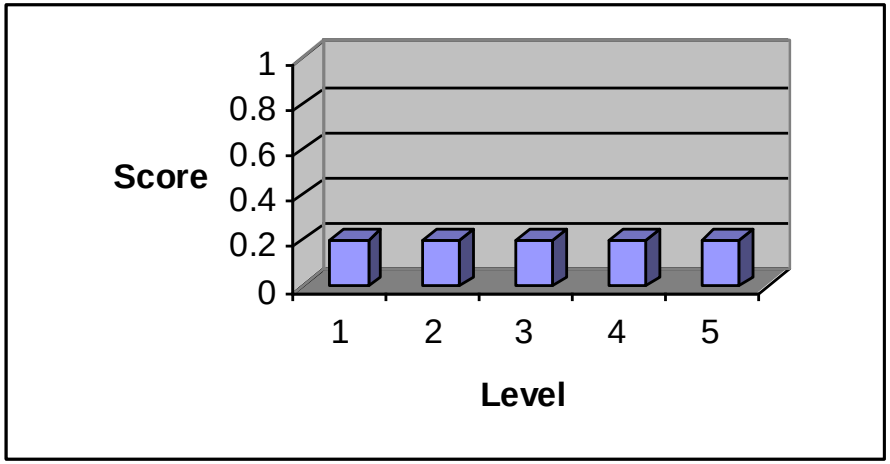
Example with $l = 5$ discrete rating levels:

1) mediocre, 2) bad, 3) average, 4) good, 5) excellent

Initial uniform reputation score before any ratings have been received.

Base rate $a(x_i) = 0.2$

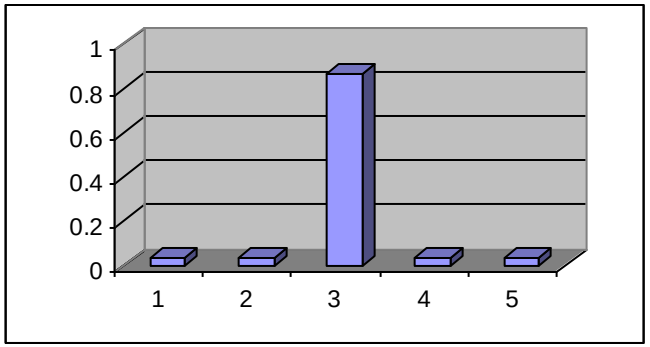
Can represent polarised ratings!



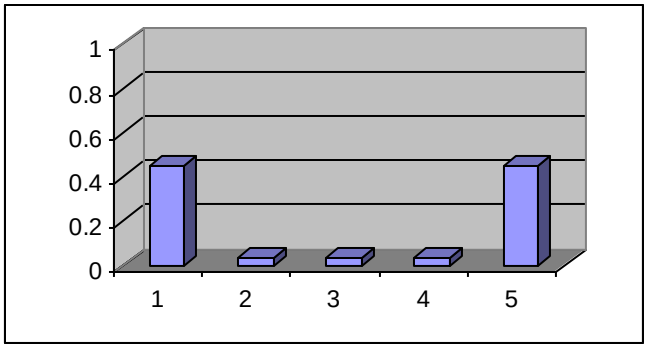
Reputation score of polarise ratings

As before, 5 discrete levels:

1) very bad, 2) bad, 3) average, 4) good, 5) very good



Non-polarised reputation score after 10 average ratings



Polarised reputation score after 5 very bad and 5 very good ratings

Computing multinomial reputation over time with fixed base rate

- $\vec{R}_i$: accumulated evidence at time i
- $\vec{r}$: evidence collected during 1 time period.
- λ : longevity factor
- $\vec{R}_i = \lambda \cdot \vec{R}_{i-1} + \vec{r}$: Recursive updating algorithm
- $\text{Score}_i(L_j \mid \vec{R}_i, \vec{a})$: Score at time period i

Score evolution over time with fixed base rate

Five discrete rating levels:

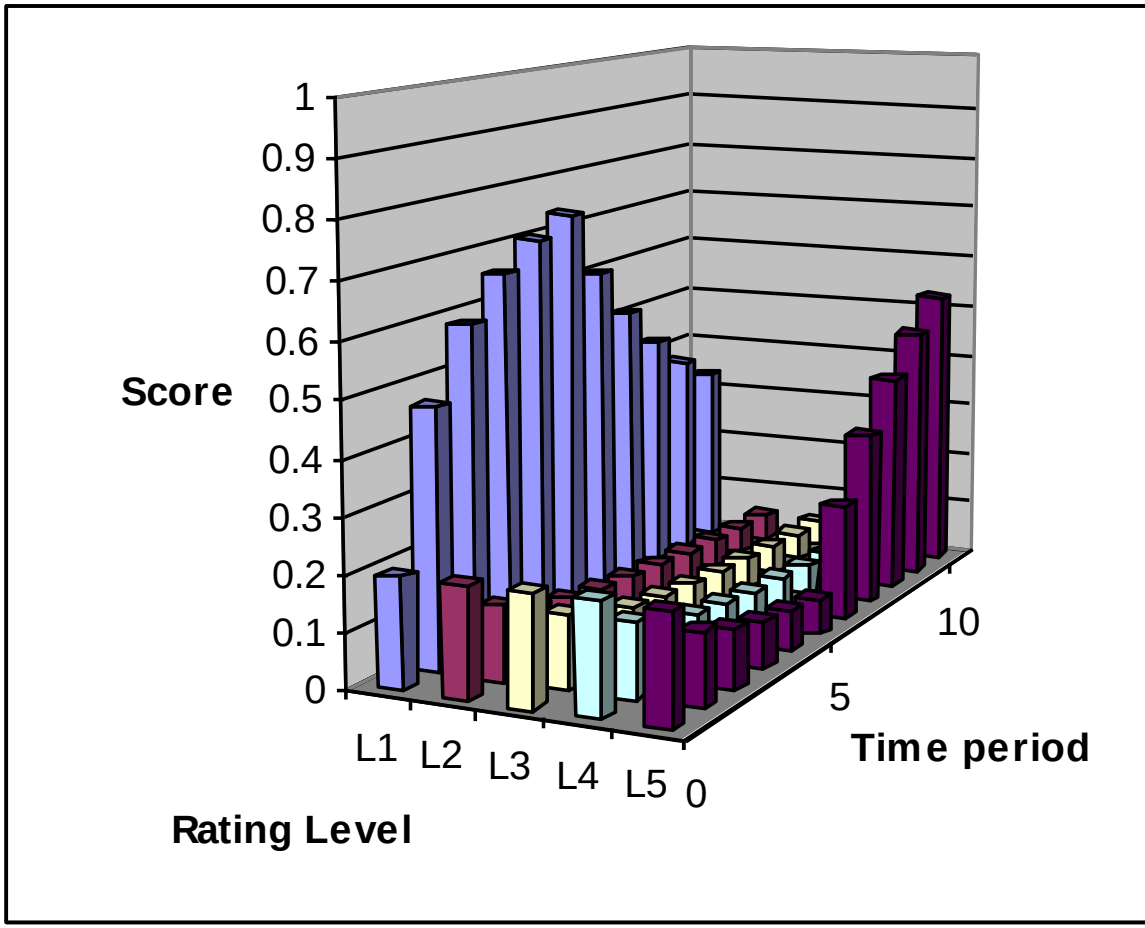
1. Mediocre
2. Bad,
3. Average,
4. Good,
5. Excellent

Longevity $\lambda = 0.9$

Base rate $a(x) = 0.2$

Periods 1-5: Mediocre

Periods 6-10: Excellent



Score evolution over time with fixed base rate

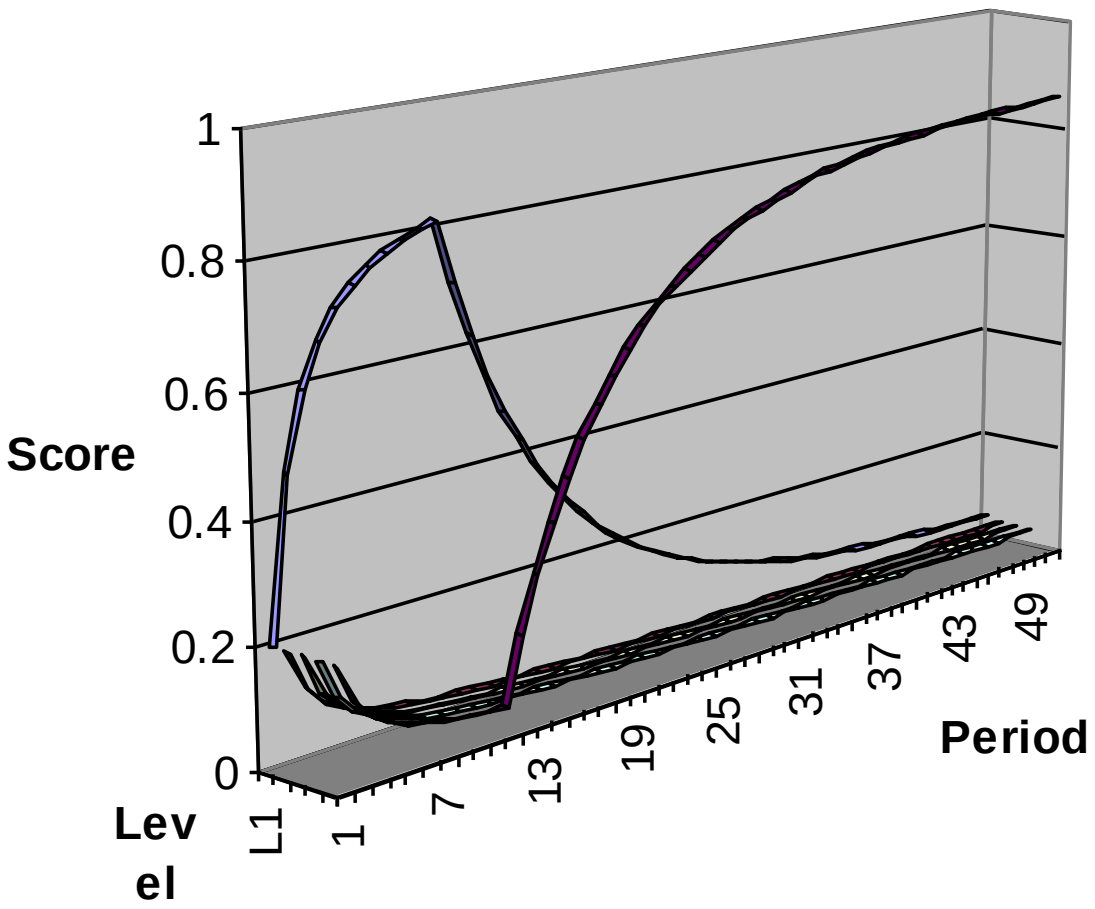
Longevity $\lambda = 0.9$

Base rate $a(x) = 0.2$

Periods 1-10: Mediocre

Periods 11-50: Excellent

The max and min reputation score is determined by the longevity factor λ



Score evolution over time with dynamic base rate

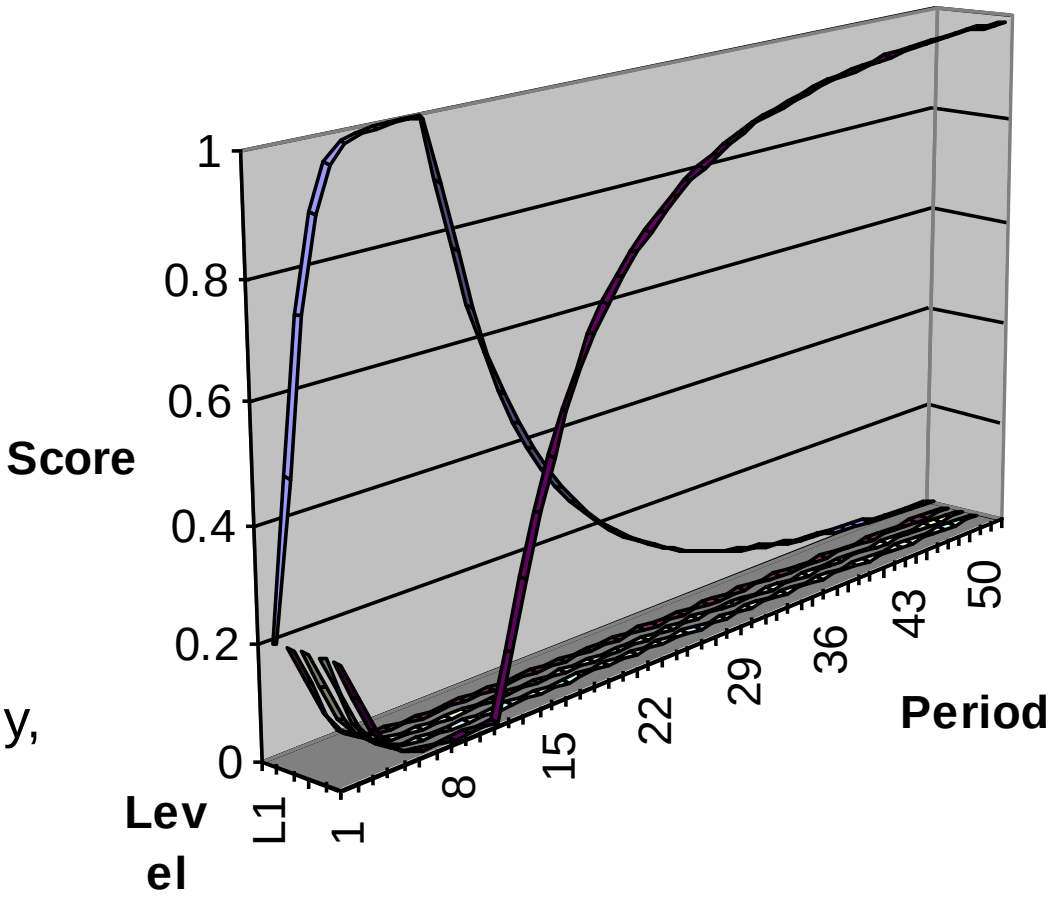
Longevity $\lambda = 0.9$

Base rate $a_{i+1}(Lj) = \mathrm{E}_i(Lj)$

Periods 1-10: Mediocre

Periods 11-50: Excellent

The max and min reputation scores are 0 and 1 respectively, and are independent of the longevity factor λ.



Point Estimate Reputation Score

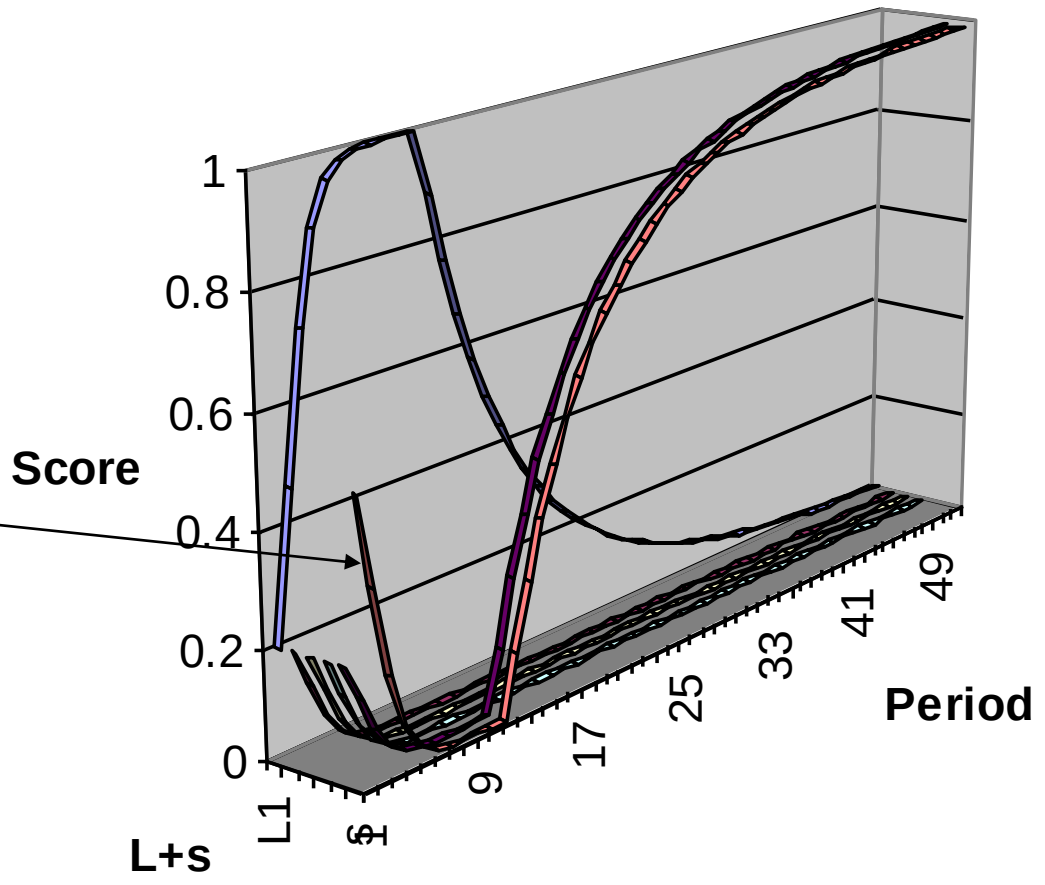
- Sometimes useful to have a single-valued score
- Translate multinomial score to point-estimate score
- l : number of different rating levels
- j : particular rating level
- $v(L_j) = \frac{j\text{-}1}{l\text{-}1}$: Point value for each rating level
- $\sigma = \sum_{j=1}^{l} v(L_j) \cdot Sc(L_j)$: Point estimate

UNIK UNIVERSITY GRADUATE CENTER

Multinomial score and point estimate with dynamic base rate

- Level values:
  - $v(L_1)=0$
  - $v(L_2)=0.25$
  - $v(L_3)=0.5$
  - $v(L_4)=0.75$
  - $v(L_5)=1$

- s=σ= point estimate
- Longevity λ= 0.9
- Base rate $a_{i+1}(L_j)= \mathrm{E}_i(L_j)$
- Periods 1-10: Mediocre
- Periods 11-50: Excellent



Score and point estimate with 5 consecutive uniform rating periods

Longevity $\lambda = 0.9$

Base rate $a_{i+1}(Lj) = \mathrm{E}_i(Lj)$

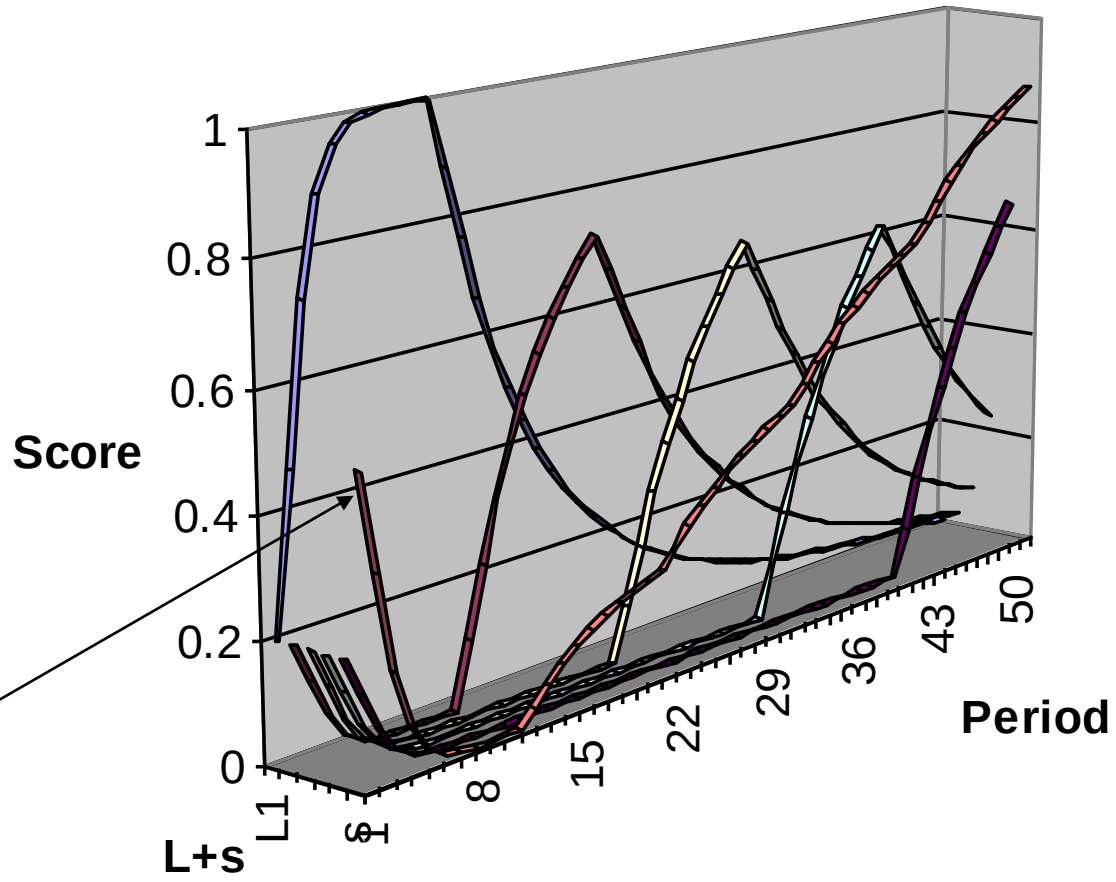
Periods 1-10: Mediocre

Periods 11-20: Bad

Periods 21-30: Medium

Periods 41-50: Excellent

•$s=\sigma$= point estimate



Discrete models

- Discrete measures
  - *“Very trustworthy”, “trustworthy”, “untrustworthy”*
- Computation
  - Heuristic formula, or lookup tables
- Simple to understand
- Qualitative
- Theoretically misguided

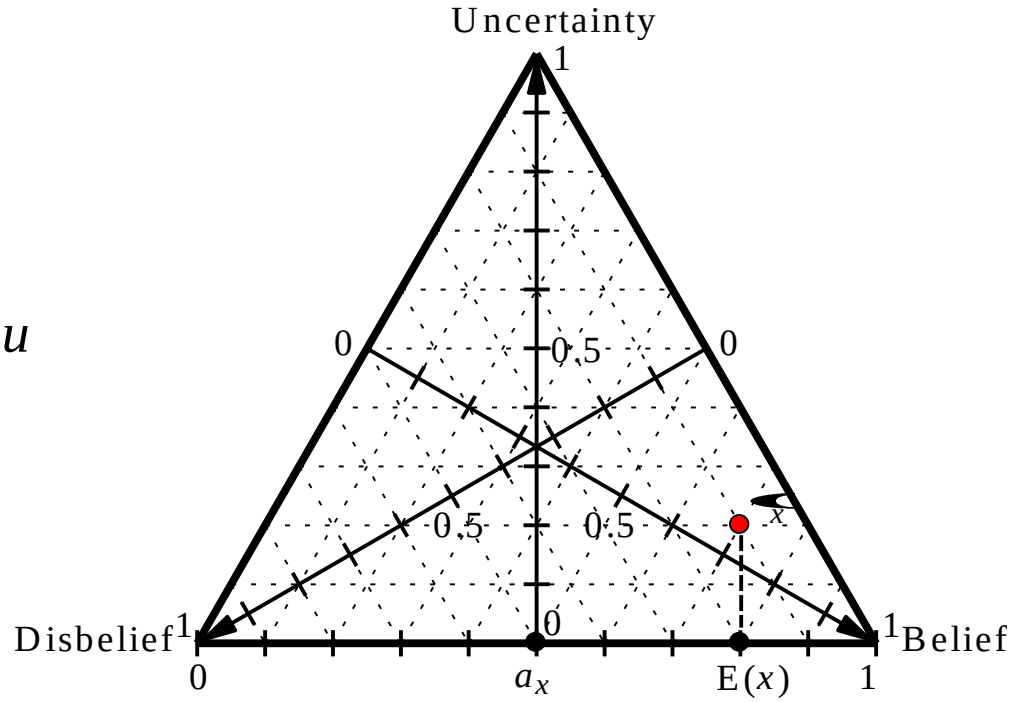
Belief models

- Assumes a trust scope σ
- Two semantic variants of each trust scope
  - Fuctional: Trust x for scope σ
    (e.g. "to be a good mechanic")
  - Referral: Trust x to refer or recommend someone/thing for scope σ
    (e.g. "to be a good at recommending mechanics)
- Two topological types
  - Direct: Trust as a result of direct experience
  - Indirect: Trust as a result of second hand evidence

Computing Trust with Subjective Logic

- Generalization of binary logic and probability calculus.
- Trust represented as binomial opinion: $\omega_x^A = (b, d, u, a)$
  - b: belief
  - d: disbelief
  - u: uncertainty
  - a: base rate

  } in range [0,1]
- Where: $b + d + u = 1$
- Expectation value: $\mathrm{E}(\omega) = b + au$
- Explicit belief ownership.



Subjective logic operators 1

Opinion operator name	Opinion operator symbol	Logic operator symbol	Logic operator name
Addition	+	$\cup$	UNION
Subtraction	-	$\setminus$	DIFFERENCE
Complement	¬	$\overline{x}$	NOT
Expectation	E(x)	n.a.	n.a.
Multiplication	·	$\wedge$	AND
Division	/	$\overline{\wedge}$	UN-AND
Comultiplication	⊔	$\vee$	OR
Codivision	$\overline{\sqcup}$	$\overline{\vee}$	UN-OR

Subjective logic operators 2

Opinion operator name	Opinion operator symbol	Logic operator symbol	Logic operator name
Discounting	$\otimes$	:	TRANSITIVITY
Consensus	$\oplus$	$\lozenge$	FUSION
Conditional deduction	$\circledcirc$	$\Vert$	DEDUCTION (Modus Ponens)
Conditional abduction	$\overline{\circledcirc}$	$\overline{\Vert}$	ABDUCTION (Modus Tollens)

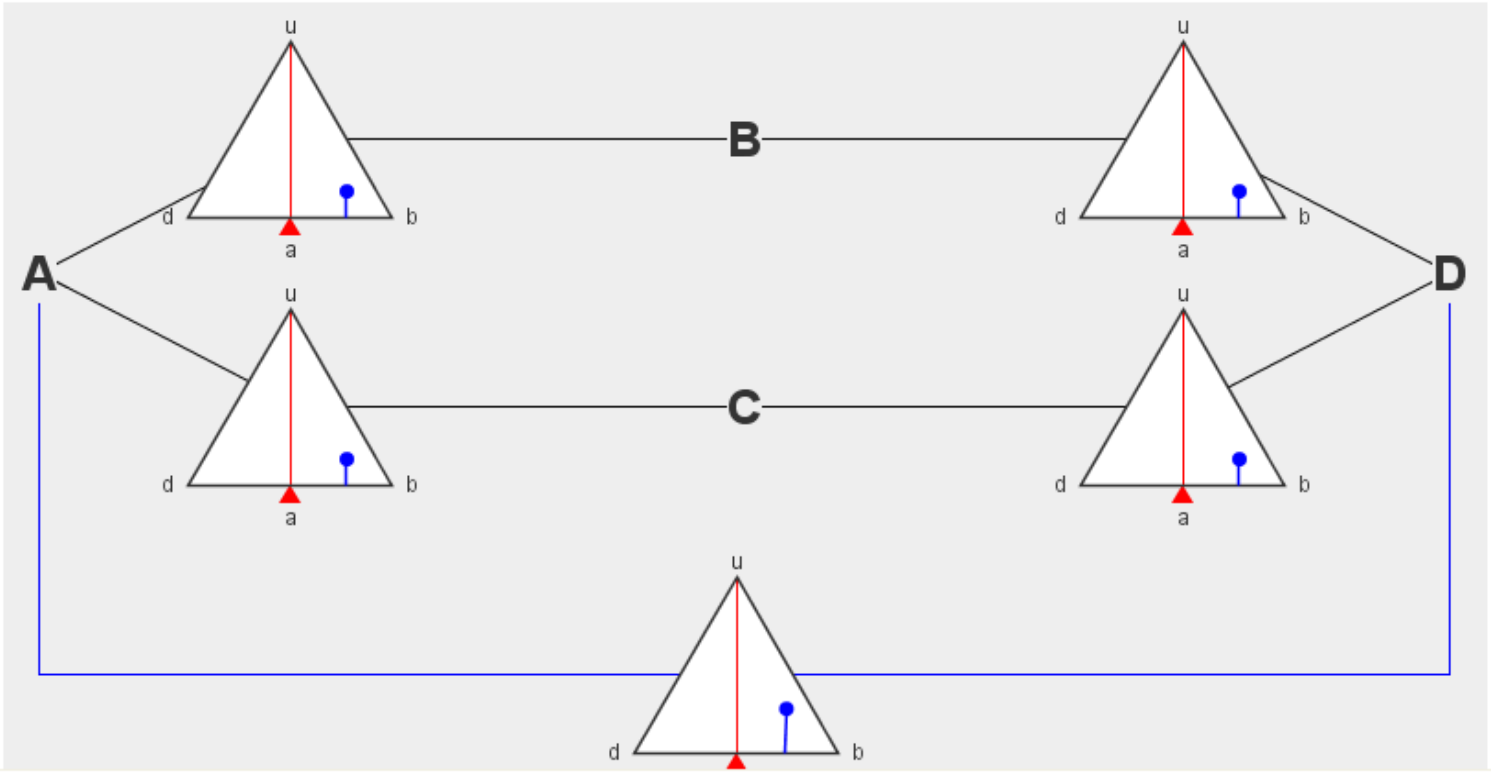
Example belief model

Simple Trust Network Demo

Four entities, labelled **A**, **B**, **C** and **D** have opinios about each other represented as points in triangles. Entity **A** is trying to form an opinion about **D**, and receives opinions from **B** and **C** as to the trustworthiness of **D**. Furthermore, **A** has his own opinions about the trustworthiness of **B** and **C**.

SPECTRUM

Left-click and drag opinion points to set opinion values. Entity **A** combines these opinions using the Subjective Logic Operators to derive his own opinion about **D**, as shown by the bottom opinion triangle. In detail, entity **A** *discounts* **B**'s opinion about **D** by his opinion about **B**, and does similarly for **C**. Finally, he combines the two discounted opinions using the *consensus* operator in order to determine his opinion about **D**. Right-click on the opinion triangles to see the exact values of each opinion. Opinion values can also be visualised using three-coloured rectangles.



Flow models

- Transitive iteration through graph
- Loops and arbitrarily long paths
- Source of trust can be distributed
  - evenly, e.g. early version of PageRank
  - discretely, e.g. current PageRank, EigenTrust
- Sum of trust can be
  - constant, e.g. PageRank
  - increasing with network size, e.g. EigenTrust

Google's PageRank

- Purpose to provide quality search results
- Based on:
  - Number of incoming links, weighted by the
  - PageRank of the sites behind incoming links
- Hyperlinks interpreted as positive ratings.
- No negative ratings.
- Random surfer model.
- PageRank is a reputation system

PageRank visualisation

- $R(A) = (1-d)/N(\text{Web}) + d \cdot \Sigma R(\text{prev}(A))/N(\text{next}(\text{prev}(A)))$
- Damping factor $d \approx 0.85$
- $\Sigma R(A) \approx 1$, i.e. $R(A)$ is the probability of the random surfer
- $\text{PageRank}(A) = l + \log_{\approx 10} R(A)$, where $l \approx 11$

Example

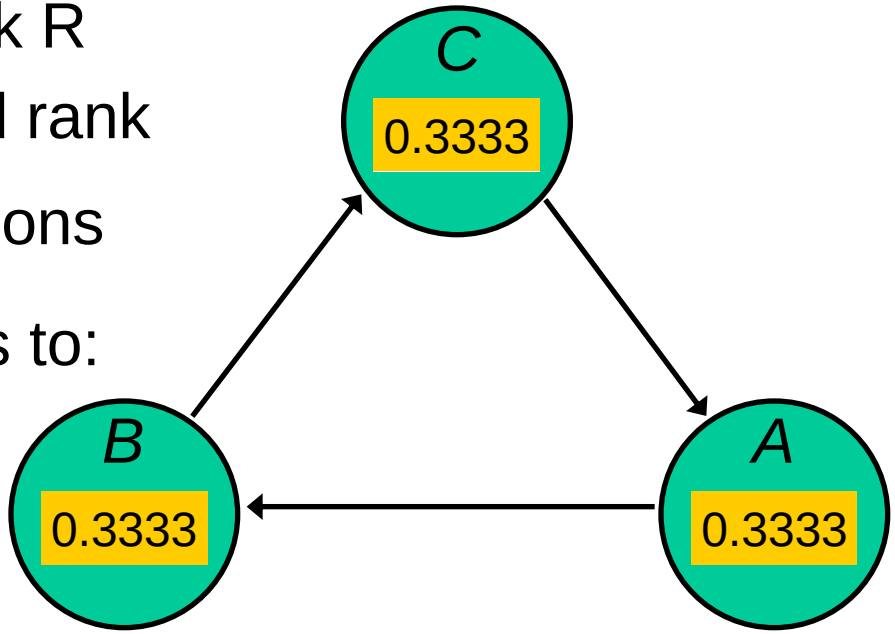
with N(Web)=3

Initial rank R

+ imported rank

and iterations

Converges to:



Link spam and “nofollow”

- Survival of e-commerce sites depends on rank
- Attempts to increase rank with link spam
  - consists of putting URLs to own Web site in wikis (publicly editable Web sites) and in postings to public discussion groups
- The “nofollow” tag, introduced in 2005, instructs Web crawlers not to follow a link

```
<a href=http://spam_site.com
 rel="nofollow">Link</a>
```

- Wikis and discussion groups now add “nofollow” to all URLs, thereby eliminating the link spam problem

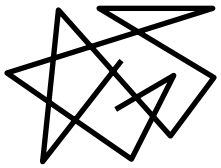
SERP Rank

- SERP: Search Engine Results Page
- SERP Rank: Position of page reference on SERP
- ≠ PageRank
- SERP Rank is a function of PageRank + constantly tuned factors:
  - Keyword position and frequency
  - Linking to good neighbourhoods
  - Freshness
  - etc.

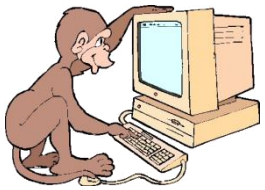
Evolution of web search ranking models

Evolution and time

Increasing quality of Web page ranking



No ranking (Altavista), ca. 1995



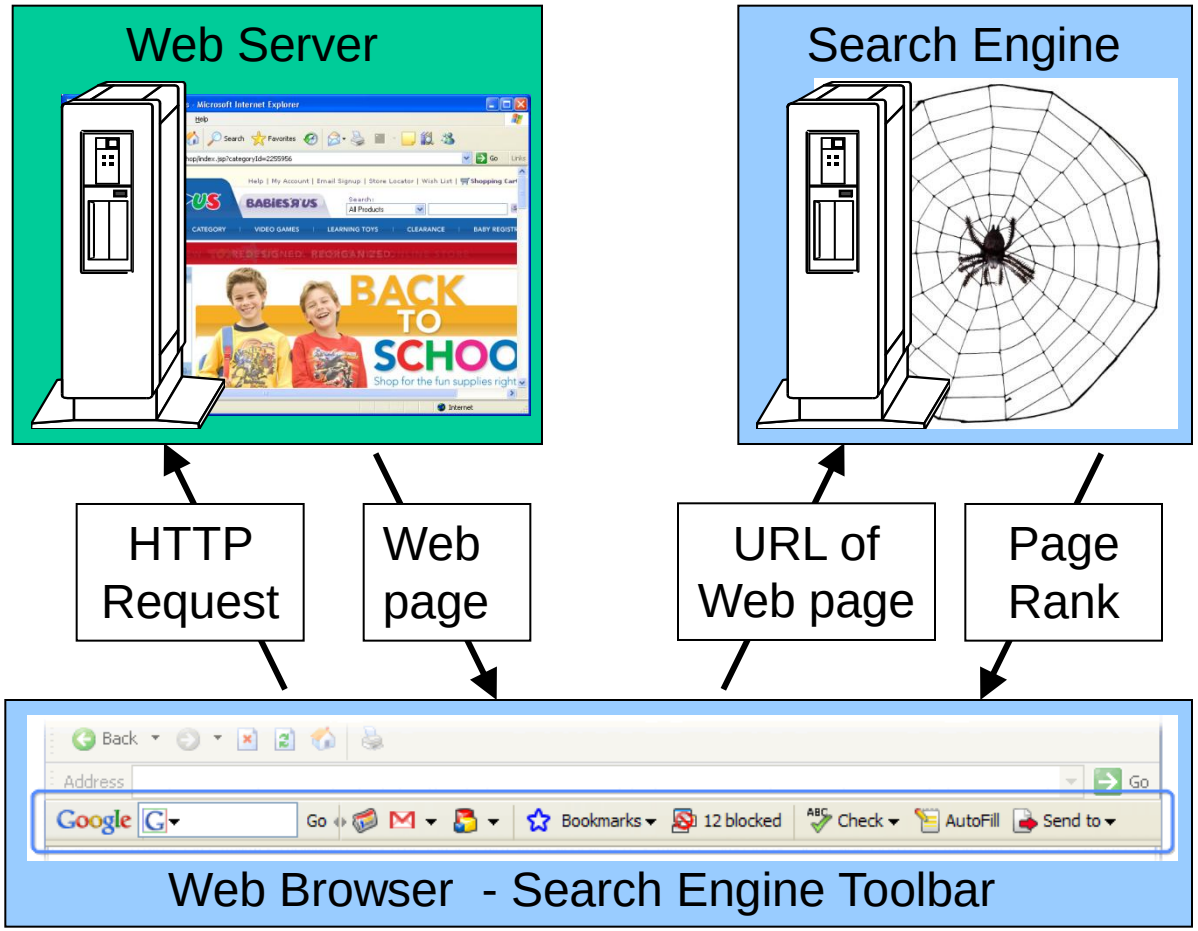
Random surfer model (Google PageRank), ca. 2000

Intentional surfer model (Google Toolbar), ca. 2005

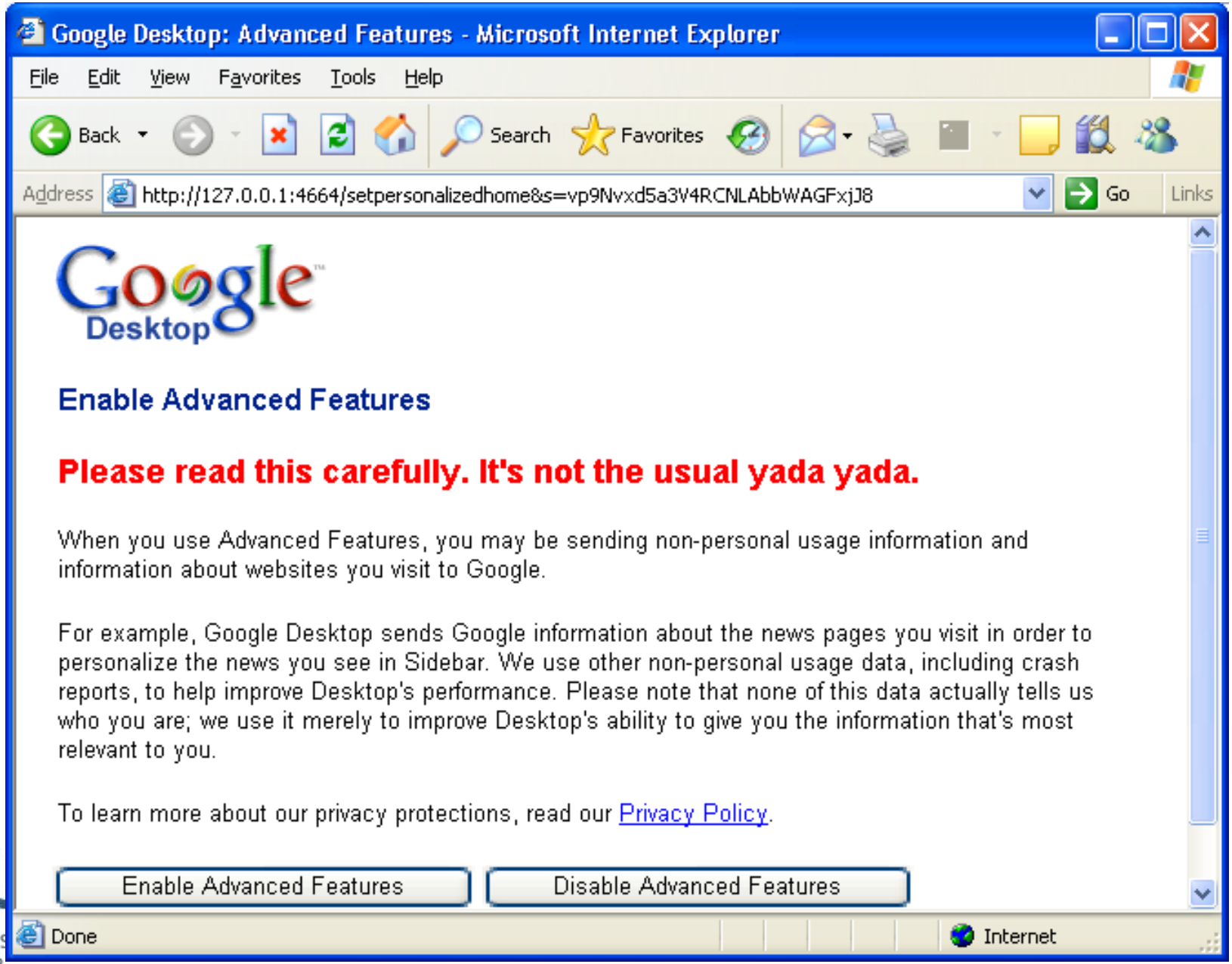
Critical surfer model (Reputation Toolbar), ca. 2010

UNIK UNIVERSITY GRADUATE CENTER

Browser toolbar architecture



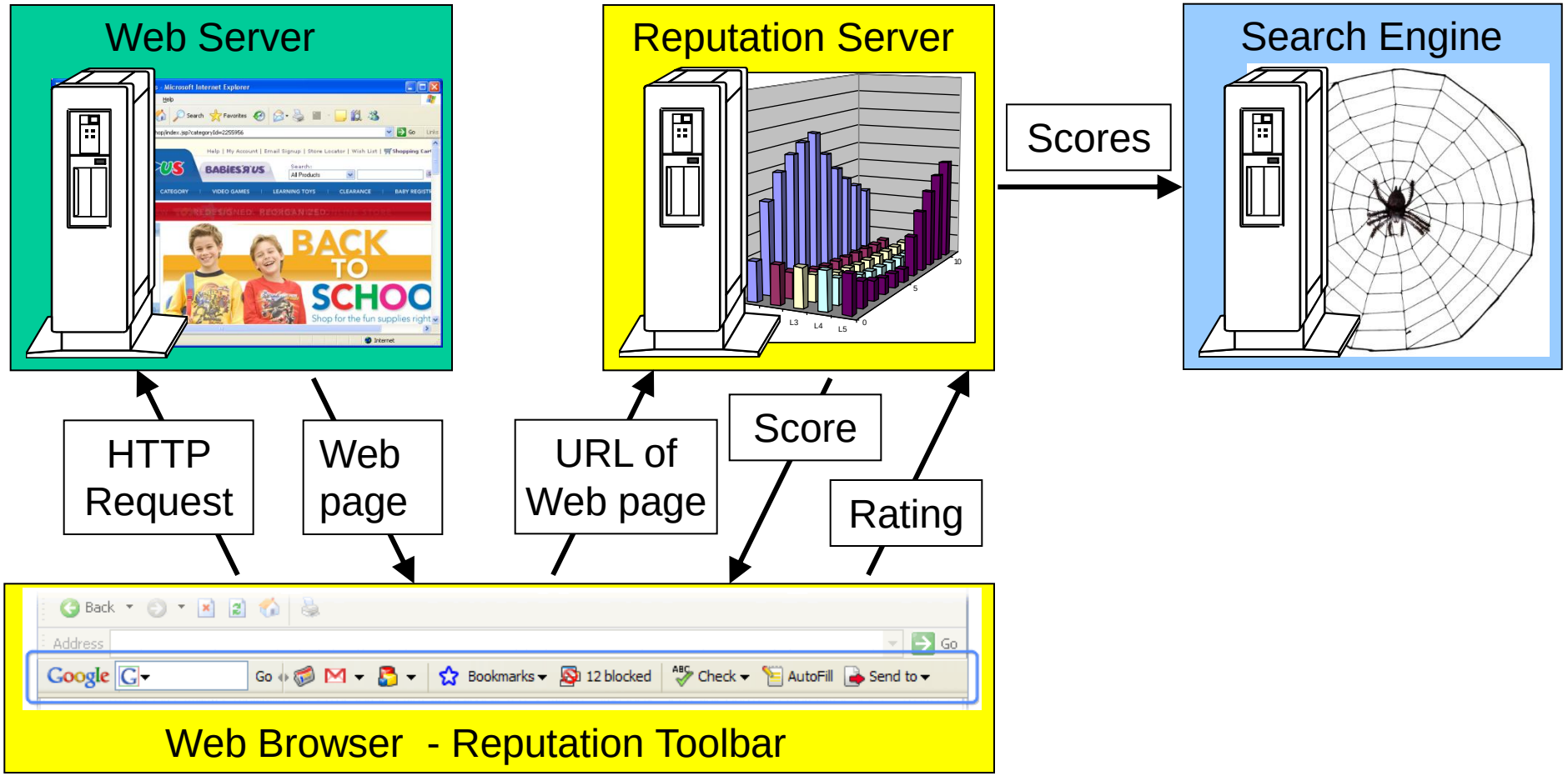
Evidence from toolbars and spyware



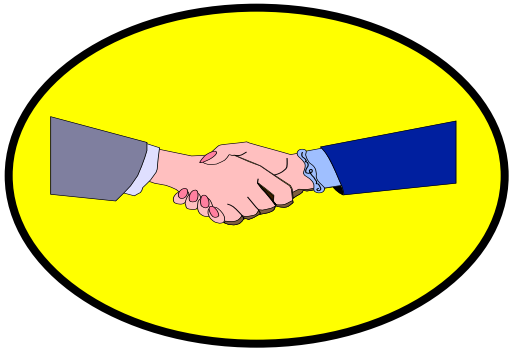
Critical surfer model

- People sometimes access a Web site even though they don't approve of its content
  - e.g. IT security researcher investigating phishing sites
- Critical surfer model depends on people rating Web pages
- Ranking = probability of people accessing a given page, weighted by its reputation score

Critical surfer model implementation



Commercial and online trust and reputation systens



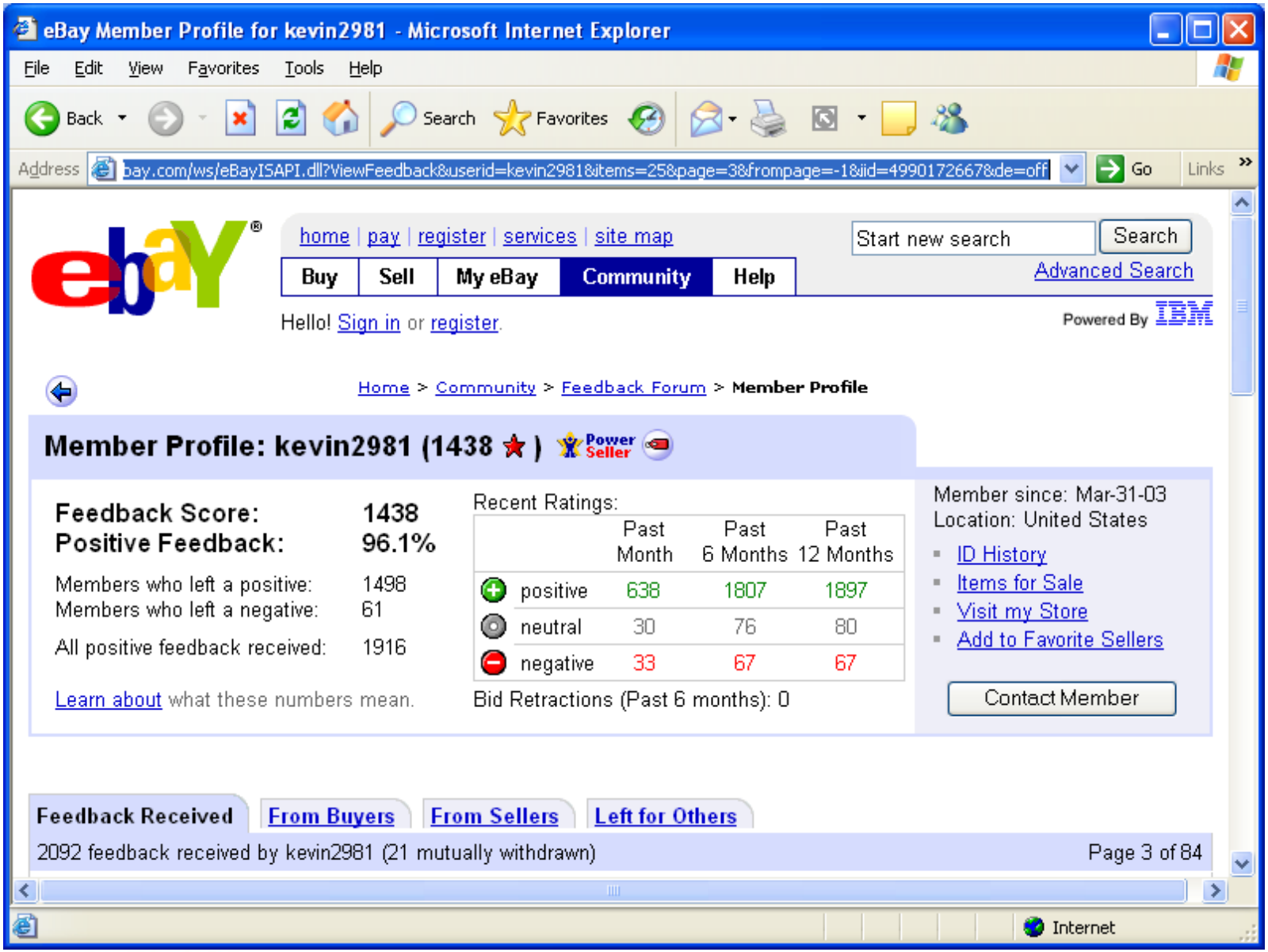
Web Sites with reputation systems

- Auction sites:
  - www.ebay.com
  - auctions.yahoo.com
- Expert sites
  - www.expertcentral.com
  - www.askme.com
  - www.allexperts.com
- Product reviews
  - www.epinions.com
  - www.amazon.com
- e-commerce
  - www.bizrate.com
  - www.virtualratings.com
- Article postings
  - www.slashdot.com
  - www.everything2.org
- Education
  - us.ratemyteachers.com
  - www.virtualratings.com
- Entertainment
  - www.citysearch.com
  - www.imdb.com
  - radio.weblogs.com

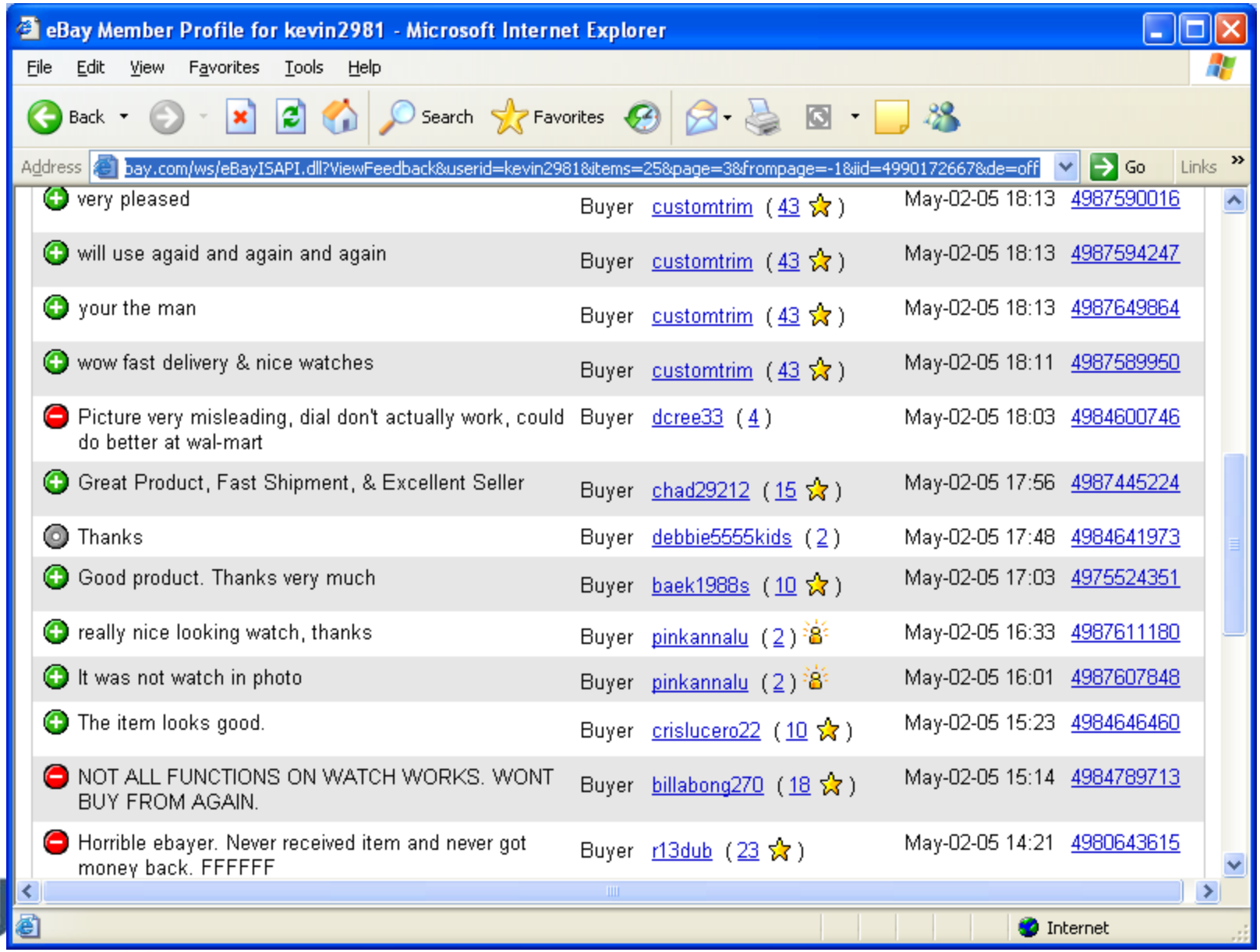
The eBay Feedback Forum

- Centralised reputation system
- Ratings:
  - Buyers and sellers rate each other, 50% - 60% times
  - positive, negative, neutral, + short comment
- Score = Σ positive - Σ negative
- Time windows
- Surprisingly positive ratings, only 1% negative
- Correlation between seller and buyer ratings
- Many empirical studies
- Purpose: to control the quality of market

Example eBay member's profile



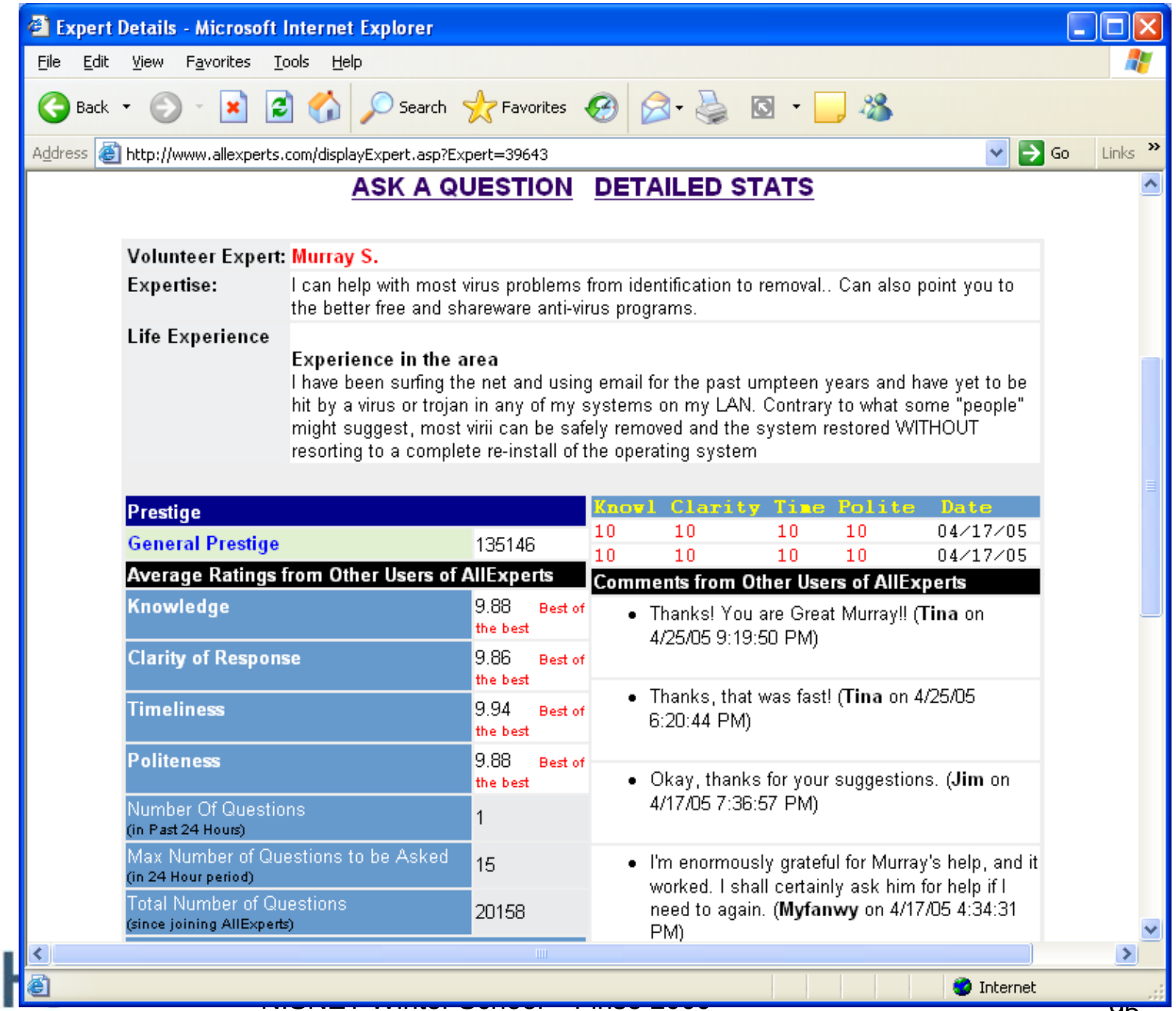
Example eBay feedback comments



AllExperts

- Free advice from volunteer experts
- Ratings given on scale [1,10] for
  - Knowledgeable, Clarity of response, Timeliness and Politeness
- Score = average of ratings
- Most experts have scores ≈ 10
- Business model:
  - Low profile advertisement
  - Prestige to volunteer experts

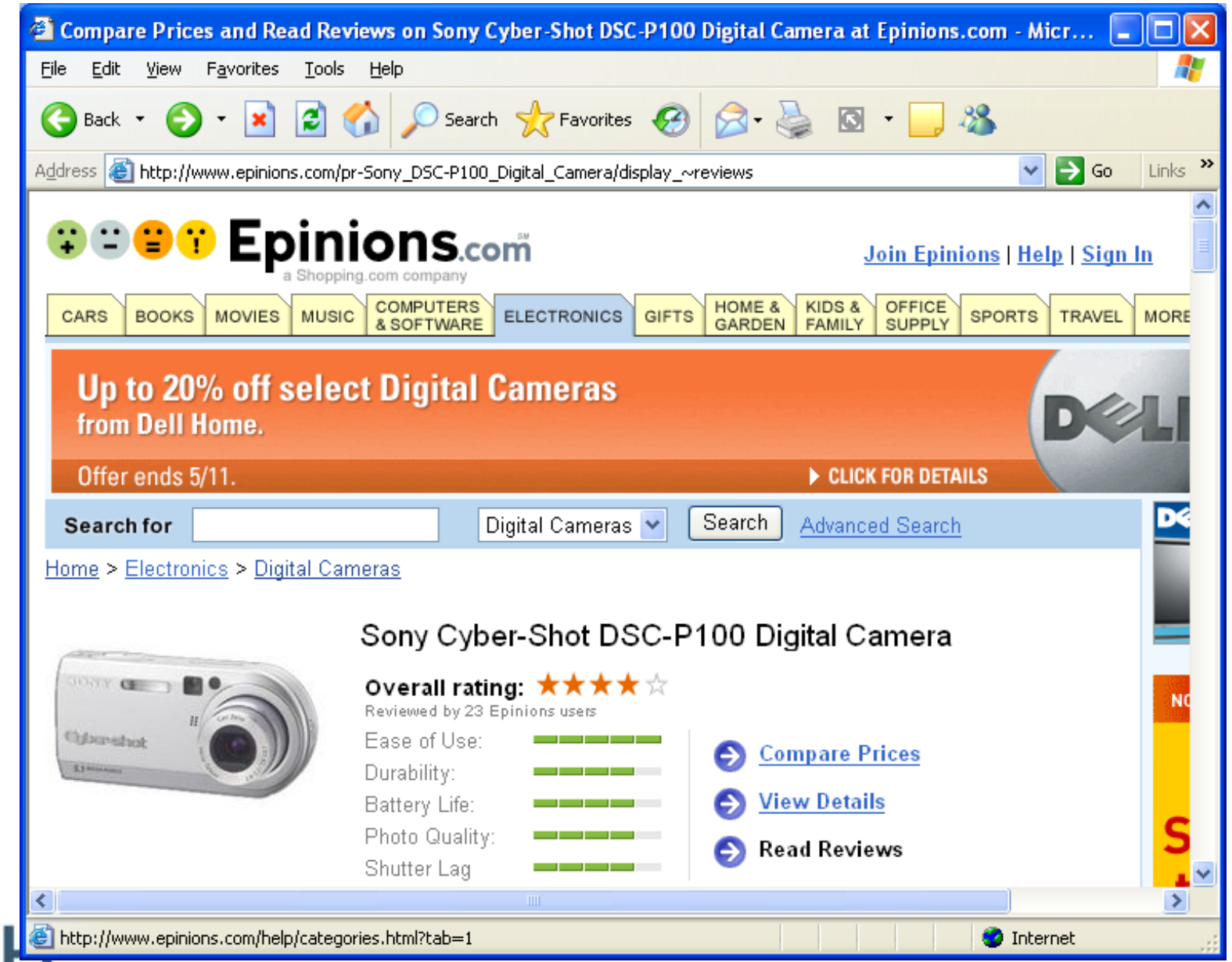
Example AllExperts profile



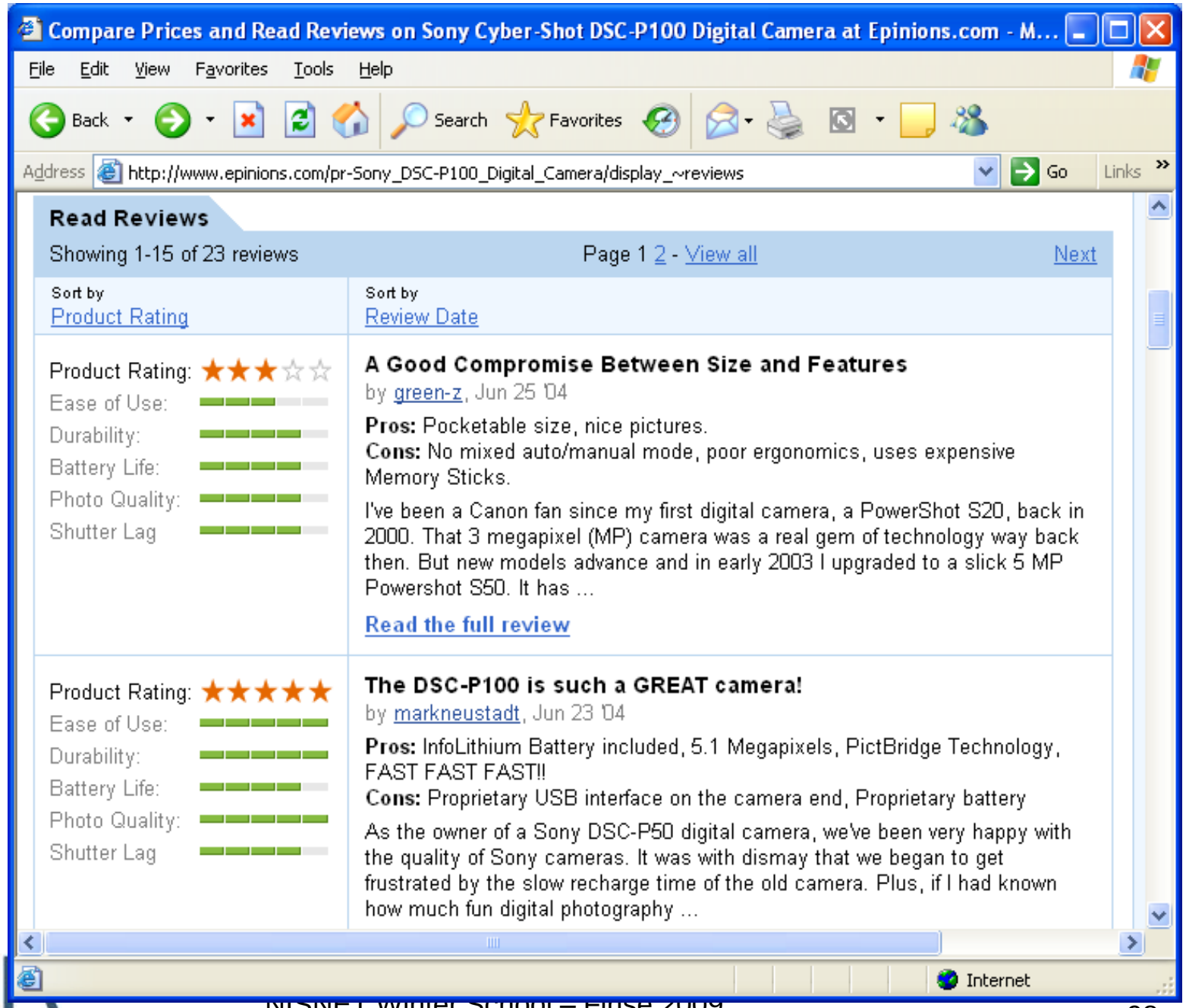
Epinions product review site

- Reviews consumer products
- Product ratings
  - in range 1 – 5 stars
  - Score = average of product ratings
- Review ratings
  - Not helpful, somewhat helpful, helpful, very helpful
  - Review score = average of review ratings
- Reviewer status
  - Member, advisor, top reviewer, category lead
- Income share program
  - Gives cash to reviewers with high number of very helpful reviews

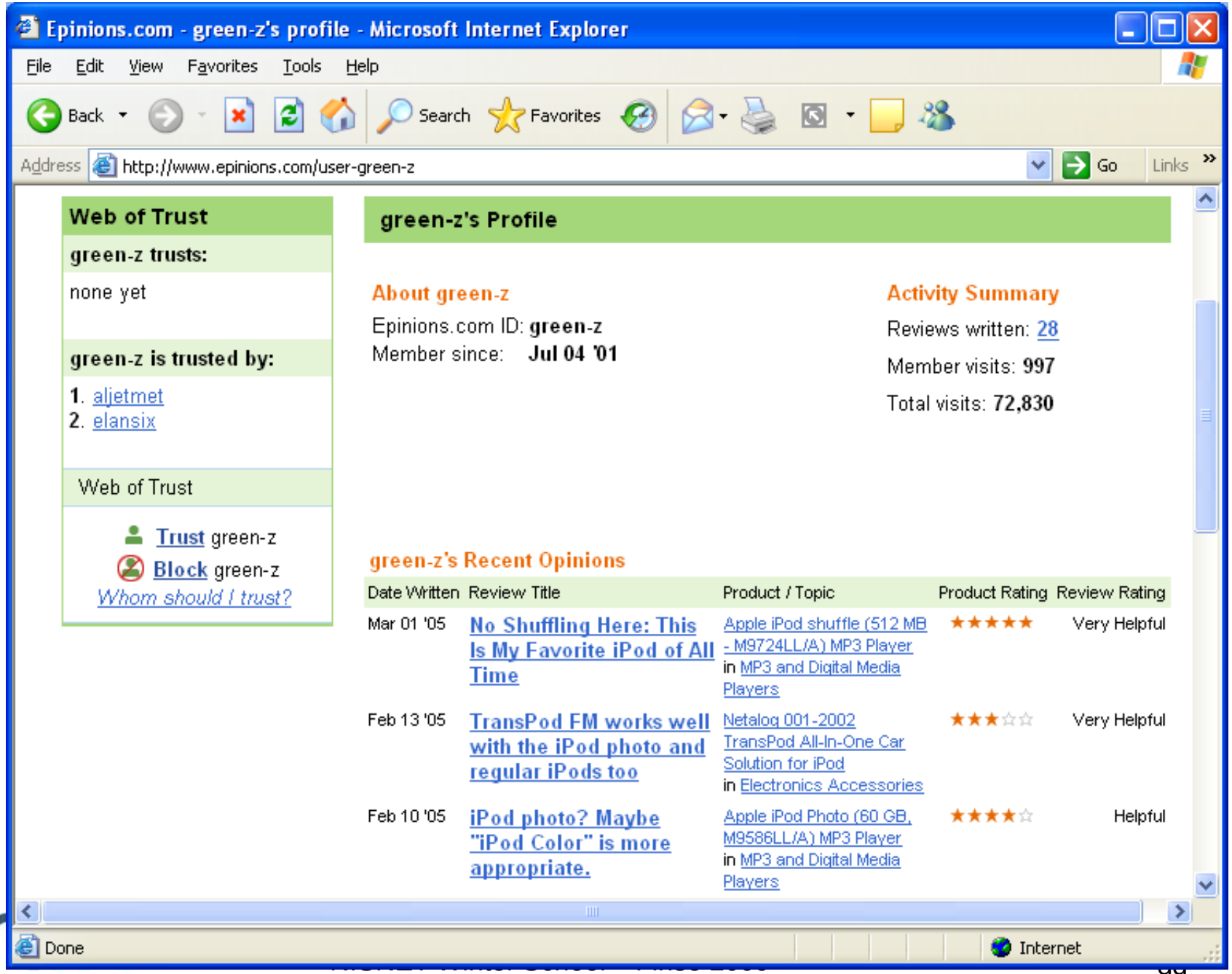
Example Epinions product profile



Example Epinions product reviews



Example Epinions member

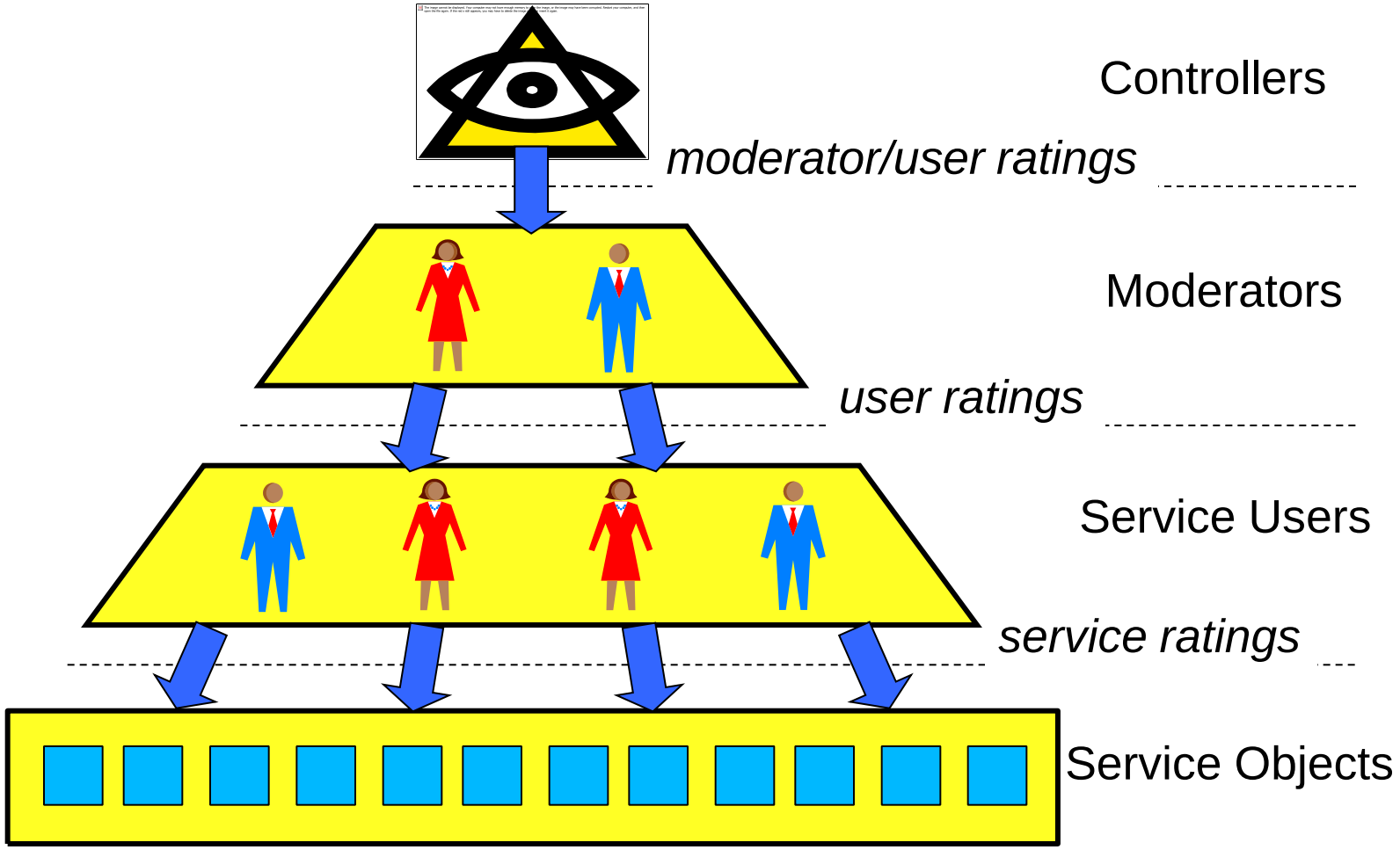


Slashdot

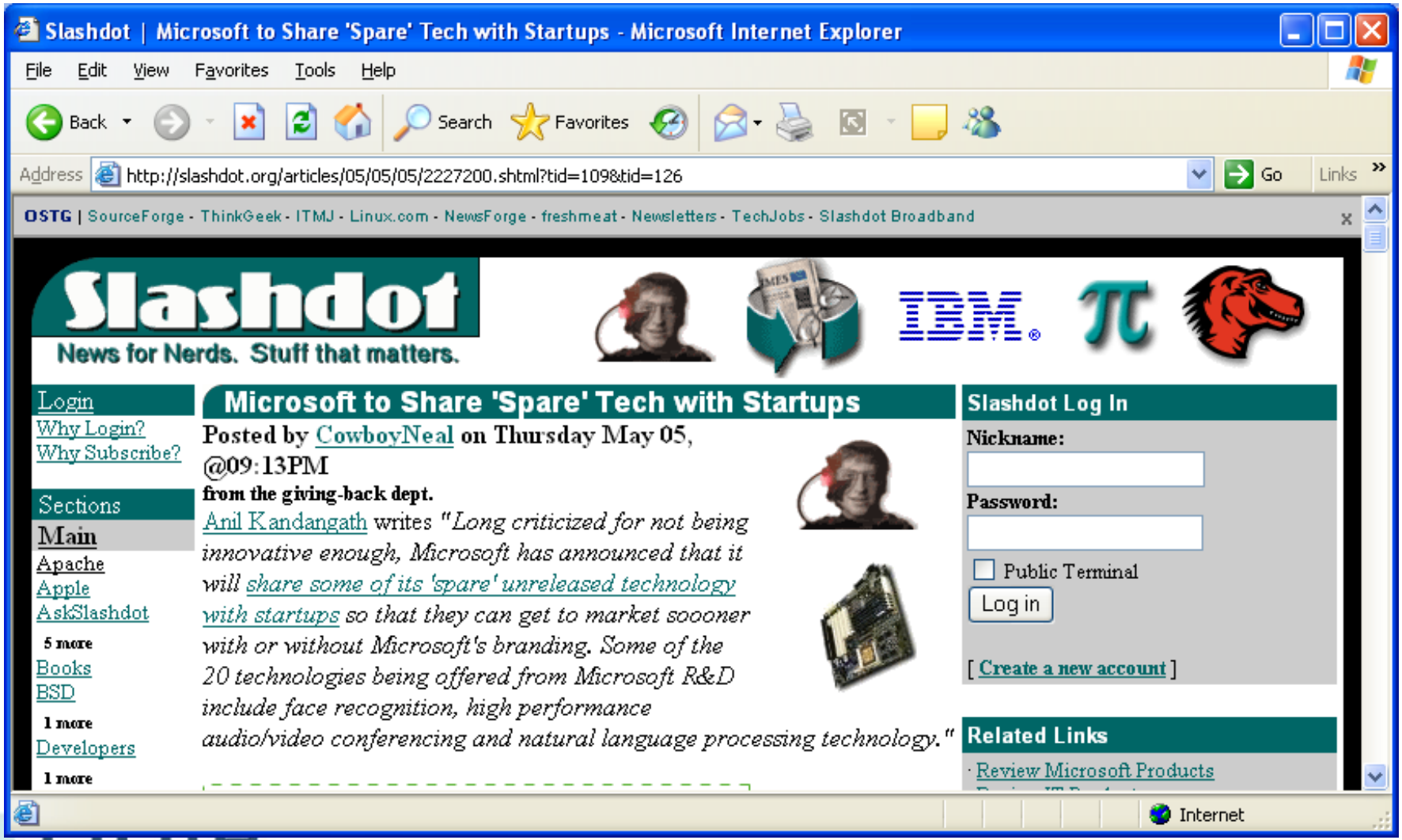
- *"News for nerds"* message board
- Article postings, at Shlasdot's discretion
- Comments to articles posted by members
- Comment moderation by members
  - Positive: insightful, interesting, informative funny, underrated
  - Negative: offtopic, flamebait, troll, redundant, overrated
  - Comment score ≈ Σ positive(Karma) - Σ negative(Karma),
  - Moderation by members with high Karma carries more weight
- Comment viewing filtered by score
- Member Karma
  - Terrible, bad, neutral, positive, good, excellent
  - Based on moderation of comments.
- Metamoderation, to combat unfair moderation
  - Rate the moderations: fair, unfair, neutral
  - Affects Karma of member who gave the moderation
- Arbitrary moderation by Shlashdot staff
- Purpose: Directing massive collaborative moderation effort

Hierarchic reputation architecture

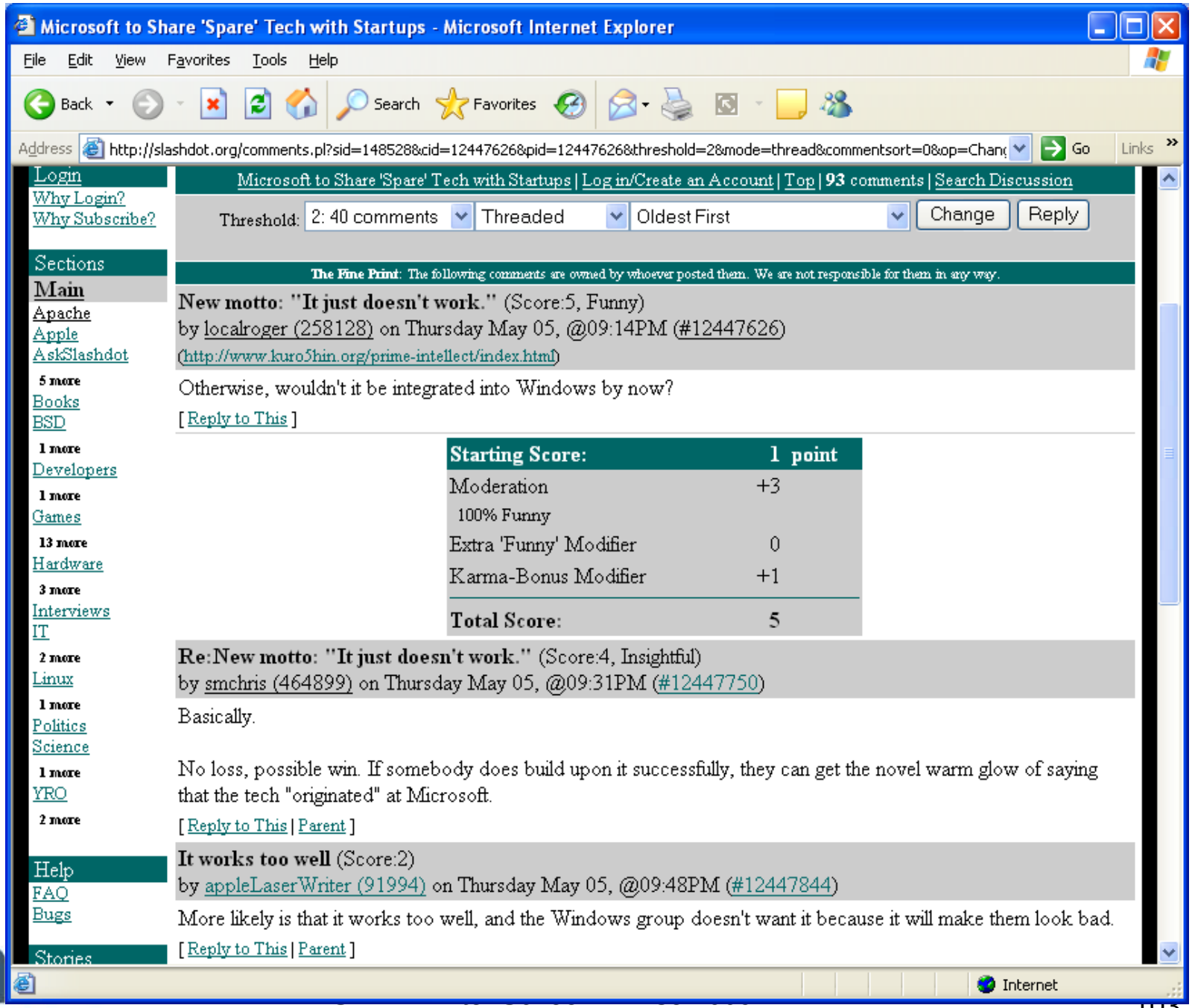
Shlashdot model



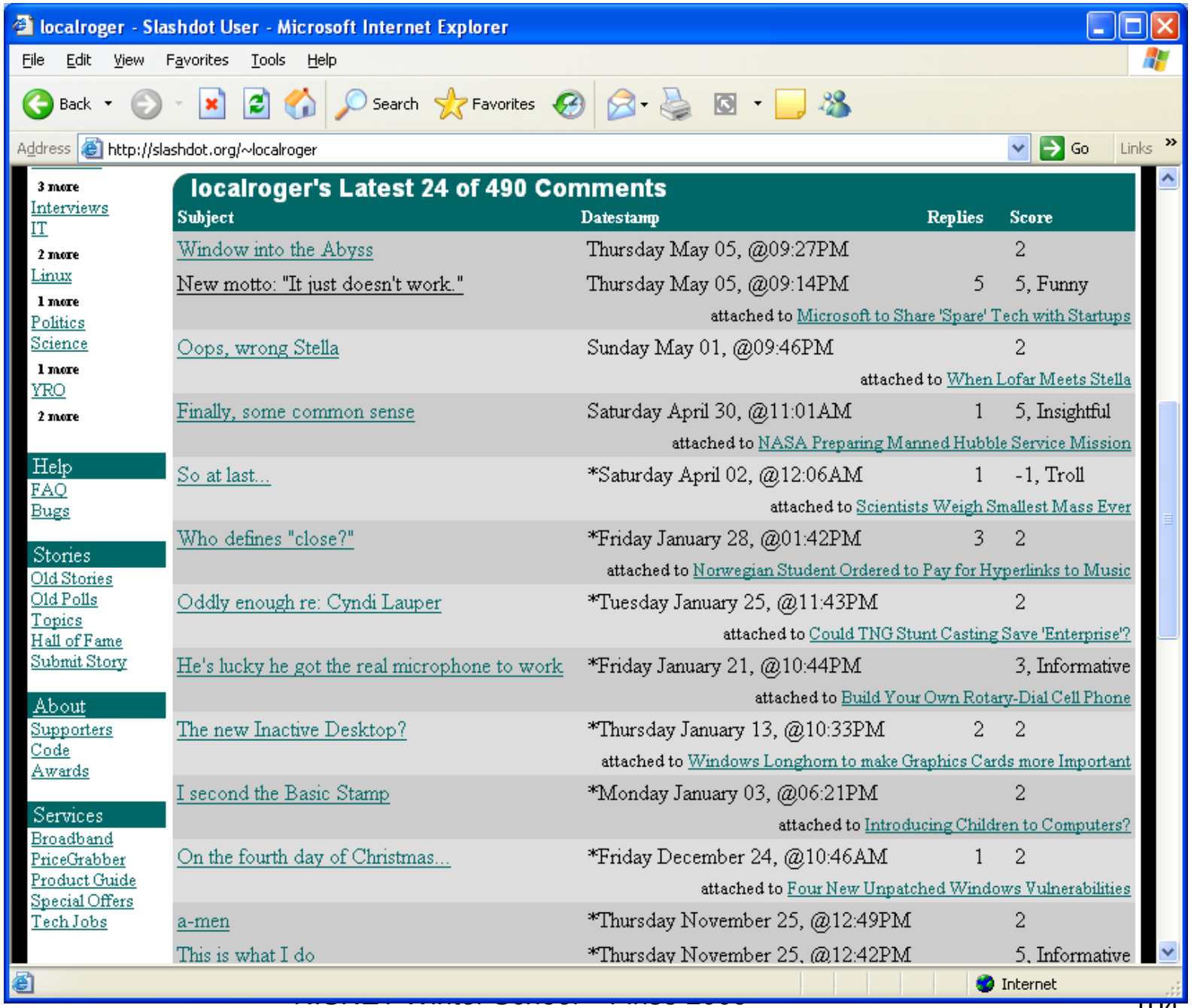
Example Slashdot posting



Example Slashdot comments



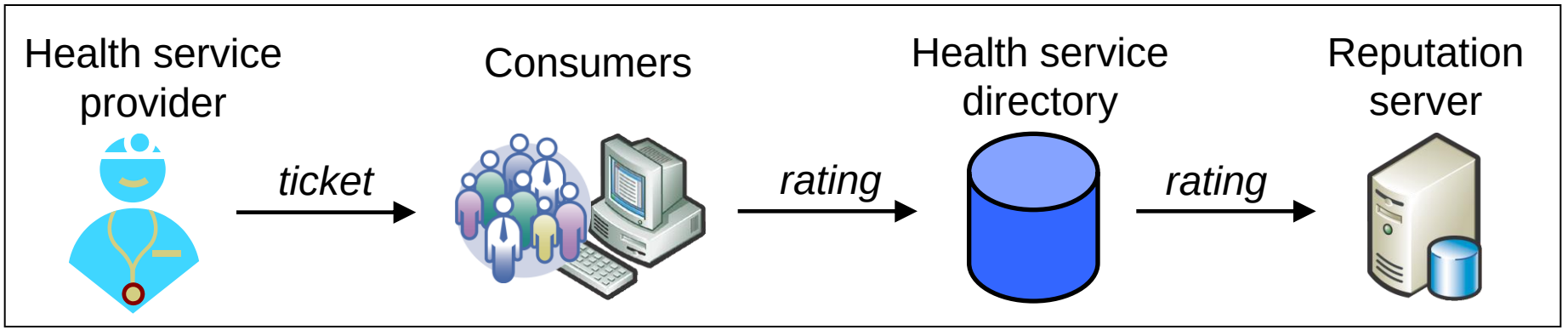
Example Slashdot member



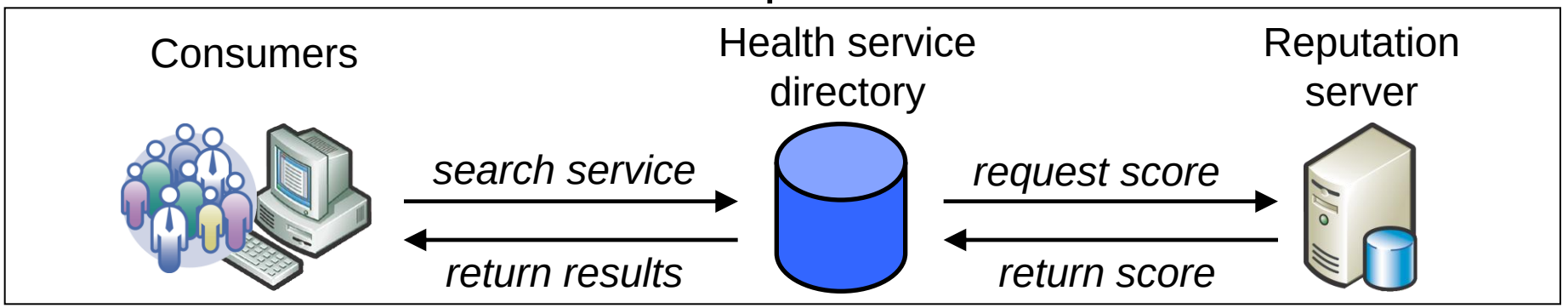
Online reputation for physical services

Architecture for health reputation system

Rating phase



Search phase



Problems and proposed solutions

Reputation System Challenges

- Ad hoc computation
- Collusion
- Unfair ratings
- Change of identity
- No incentive to provide ratings
- Hard to elicit negative feedback
- Discrimination

Unfair/False Ratings

- Amazon.com revealed true identities of reviewers by mistake in 2004
  - Reviews & ratings were written by authors/publishers/competitors
- Political campaigns promote positive and hide negative video clips on YouTube by unfair ratings
  - Use programs that mimic legitimate YouTube traffic
  - Botnets are probably being used
- eBay users are buying and selling feedback

What about subjective taste?

- Collaborative Filtering System
  - Assumes different taste
  - Identifies like-minded with same taste
  - Recommender systems
- Reputation System
  - Assumes consistent quality judgement
  - Sanctions poor quality
  - “Collaborative Sanctioning System”

Yhprum's Law

(systems that shouldn't work sometimes do)

- People provide ratings despite having no rational incentive to do so.
- Negative ratings are hard to elicit.
- Relatively easy to mount attacks against existing reputation systems.

- A reputation system works when people can relate to it
- Supports community building

Countermeasures against attacks

- Sound computation engines
- Authentication/security
  - Prevents change of identity
- Statistical filtering, and discounting
  - To prevent unfair ratings, discrimination and collusion
- Anonymity
  - To prevent fear of retaliation
- Benefits / special offers
  - To provide incentive

Concluding remarks

- Commercial online systems use very primitive computation engines
  - It is important that users can relate to the systems
  - Community building is an important factor, in addition to enhancing market quality
- Many different proposed theoretic systems
  - Little coherence among researchers
  - Pioneering period
  - No one system is optimal in for all applications
- Challenging to make systems robust against attacks
- Potential for systems that
  - Increase the quality of online markets and communities
  - Provide incentive for good behaviour
  - Complement traditional security mechanisms