A fully distributed recommendation system for pervasive devices

Seamus Moloney, Nokia Research Center, Helsinki

Nokia is about Connecting People

Pervasive computing *requires* device-to-device opportunistic interactions between strangers.

Can we improve the quality and reliability of these interactions by borrowing from a model which has worked pretty well on the internet for e.g. EBay? Are trust engines feasible for mobile devices?

Outline

- Pervasive networks, trust engines illustrated
- User perception of a trust engine
- Middleware and services to applications
- Example recommendation aware applications developed
- Performance assessment: footprint and simulation
- Conclusions and future work

Pervasive interactions: network characteristics

Mobile code based systems where components are downloaded locally.

Seamless content/sensor data collection dissemination via mobiles.

Finding a secure printer in an airport WLAN hotspot.

- Short Lived
- Often single hop
- Unpredictable
- Proximity based
- Opportunistic
- Potentially risky and wasteful of limited resources

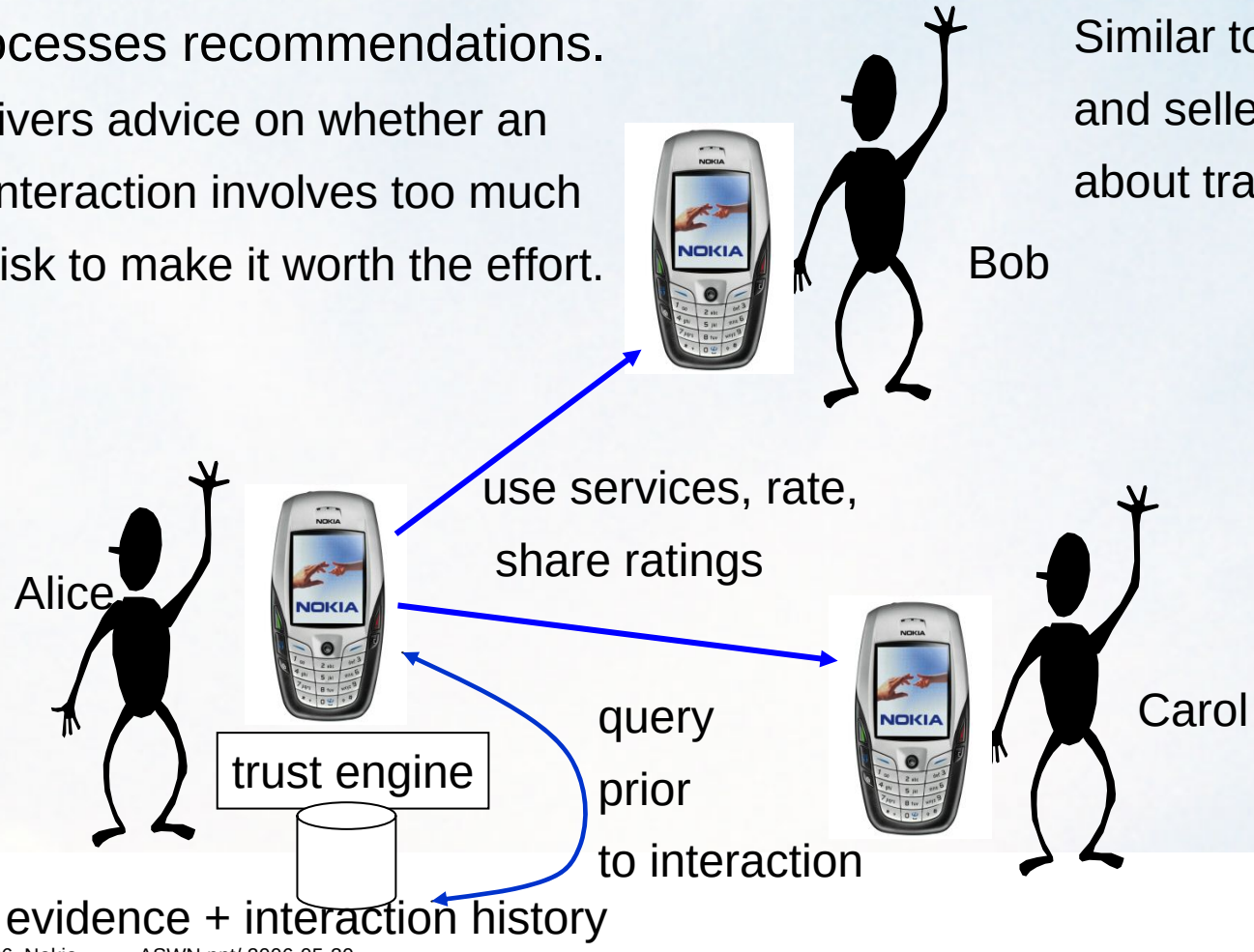
NOKIA
Connecting People

How Trust Engines operate

Assist the user in making interaction decisions:

- Gathers evidence/interaction history records.
- Processes recommendations.
- Delivers advice on whether an interaction involves too much risk to make it worth the effort.

Similar to how buyers and sellers record feedback about trades on EBay



User perception

Ebay reputation system has provided rich research material:

- not just +ve/neutal/-ve rating should be allowed
- various aspects of a trade should be ratable
- text comments are very useful for end users

Studies suggest users *want* to understand automatic decisions:

Black box trust engines are unlikely to be acceptable.

→ Developed trust engine provides transparency

- Trust database can be browsed and managed by end users
- Recommendations can be annotated & categorized flexibily
- Simple UI for viewing ratings quickly.

Users must accept the whole idea or they won't rate things

NOKIA
Connecting People

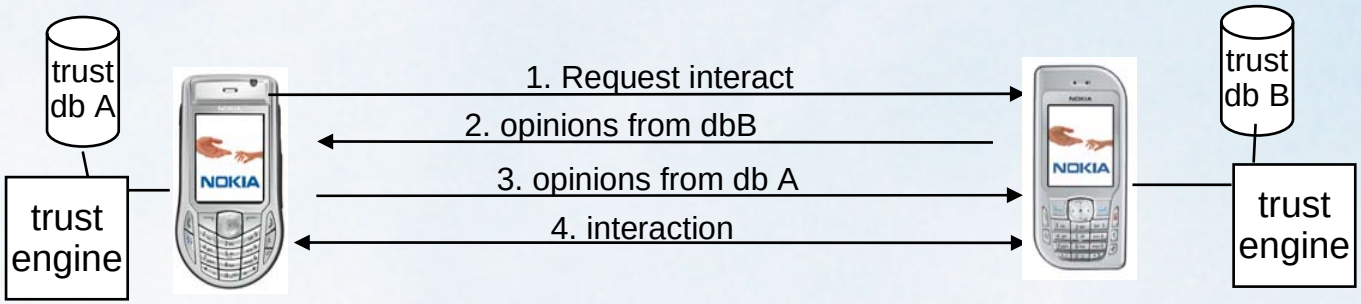
Trust Engine Features

- Decentralized storage: each device has its own collection of ‘evidence’ in the form of recommendations.
- Recommendations exchanged in XML format, can also be downloaded from e.g. the Internet.
- API allows applications to query e.g.
  - “Is it wise to proceed in an interaction with peer XYZ involving picture sharing?”
- SQL DB used for storage of recommendations → Allows engine to execute pretty complicated processing on gathered evidence when such queries are made.

→ Personalized decision making at each device, device owners own ratings have significant impact

Applications built on the engine: PicShare for proximity

Sharing Directly over proximity: share ratings gathered



Peers download pictures from those in proximity.

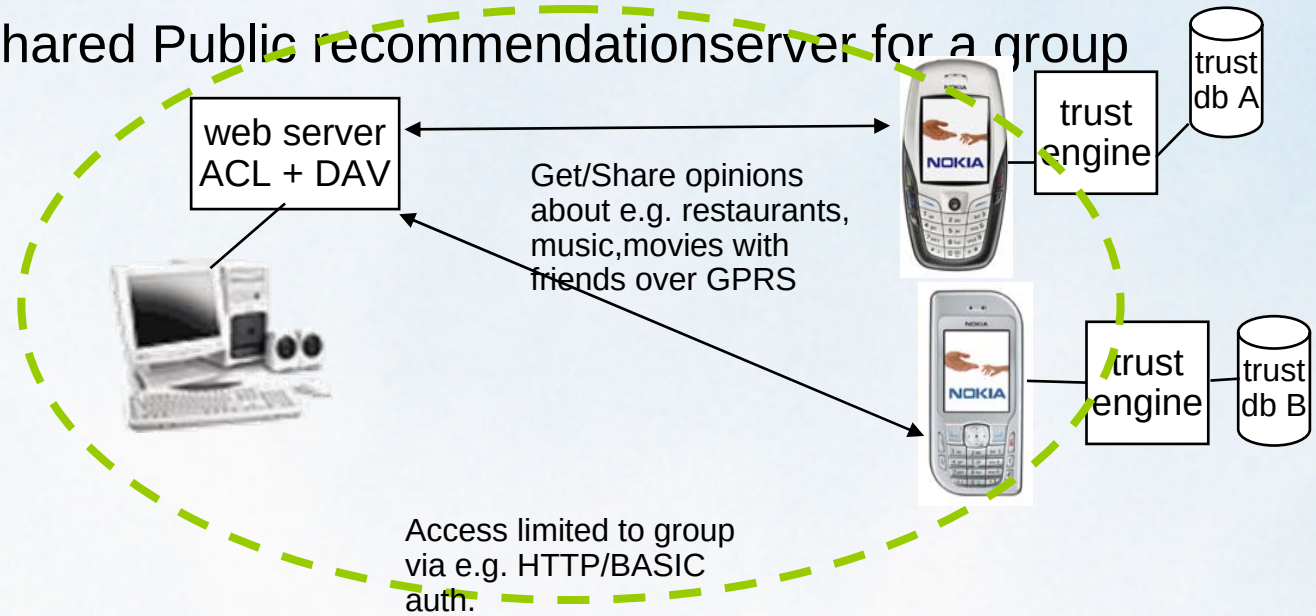
Each interaction can be rated and the ratings are shared

Query the trust engine before each interaction

Share gathered ratings as part of the interaction (piggyback)

Applications built on the engine: TrustNet

1. Shared Public recommendationserver for a group



Intended as a way for groups to share location specific recommendations e.g. about restaurants in a city.

Recommendations shared as XML documents over GPRS via WebDAV

Enables use cases like “suggest me a restaurant within walking distance..”

NOKIA Connecting People

Trust Model implemented principles

- Applied accepted principles from the trust management literature :
- When an application makes a query to trust engine, apply rules:

  From EU IST Secure results:
  - Value self-created recommendations most highly
  - Accept that some identifiers are stronger than others and reward those
  - Allow for a risk profile to be configured from the API

  From Liu,Issarny at iTrust 2004
  - Keep track of whose ratings work best for you (credibility)
  - Apply time-fading of ratings to favor the fresh ones

  From Kinateder,Rothermel at iTrust 2003
  - Fully matching results to queries weighted highest, but partially mathcing used

  →**Should be a pretty robust, realistic trust engine**

  →**But does it work in practice???**

NOKIA Connecting People

Physical Limitations: results from Nokia 6600 running the trust engine

- Avoid centralisation : Each device needs to keep its own DB.
- Import/Export == time to form or parse 20 recommendations for other party, conversion form DB format to XML, vice versa
- Calculate time: process all stored evidence and make a decision: allow interaction?

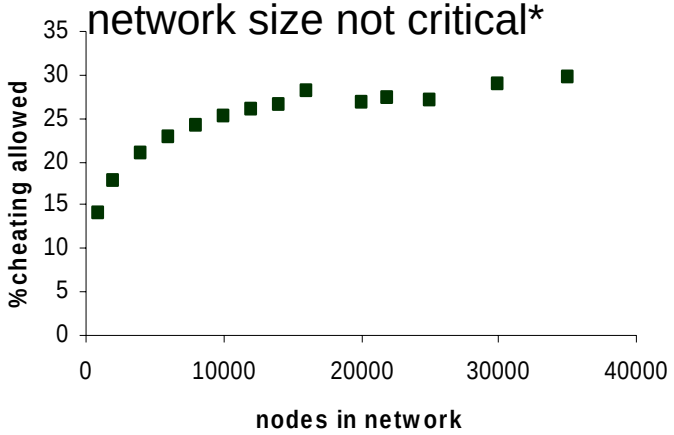
#Recommendations	100	500	1050	1830	2885
Calculate time (ms)	140	343	515	968	1075
Export time (ms)	109	312	468	703	1078
Import time (ms)	984	1015	1197	1250	1375
Device DB size (kBytes)	9.2	16.5	80	153	218

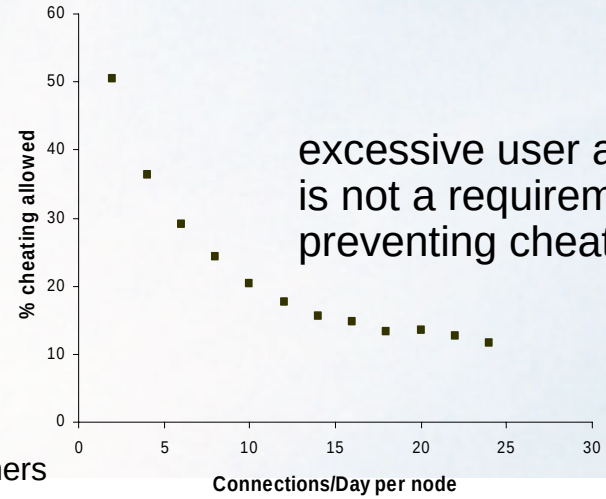
- Engine size: < 100kBytes, native Symbian implementation
- Critical limit to storage in 6600 was ~3000 recommendations stored

Simulating trust engine usage: Feasibility for proximity security

- Assuming user movements can be mapped as a social network, can use small world modelling tools to model the way pervasive devices interact.
- Modified a Watts-Strogatz small world model to give each node a ”recommendation DB” and to make trust decision prior to interactions
- We applied the memory and data exchange restrictions to this model and found:

recommendation-based protection: network size not critical*





excessive user activity is not a requirement for preventing cheating*

*Graphs show how nodes e.g. sharing bad MP3 are avoided by others once word of this spreads via recommendation sharing: simulation using small world modelling

Some problems identified

- Semantics of what it is that is being rated/recommended:
  - Ontologies add accuracy, allow agreement on naming, but a loose “tagging” approach might be more practical.
  - Combining content quality measurements and security information can result in too generic a UI.
- Varying degrees of peer recognition:
  - Strong identifiers (PKI based) allow
    - prevention of recommendation forgery
    - more accuracy when using recommendations
    - tracking.. i.e. they sacrifice privacy
  - But weaker identifiers like MAC address are more likely to be dominant
    - Trust Engine needs to be able to recognize this difference in quality of identifier!
- Chicken and egg problem is only resolvable by making it really intuitive for users to create recommendations
- Dissemination of data can be piggybacked but this needs optimization

Conclusion and future work

- Recommendation systems represent one lightweight way of improving the security and enjoy ability of the “free-for-all” nature of pervasive interactions between devices.
- These systems should be as transparent as possible to get user acceptance.
- Current smart-phones have the processing power and capabilities to support a sophisticated trust engine.
- Performance and simulation have shown the “gossip spreading” model works pretty well and is feasible.
- Keeping the engine free of the transport details enables good flexibility for different use cases.
- TODO s : investigate integrating to p2p reputation systems like Credence (for Limewire), becoming part of a more general purpose firewall.

Questions?

research.nokia.com

Thank you!