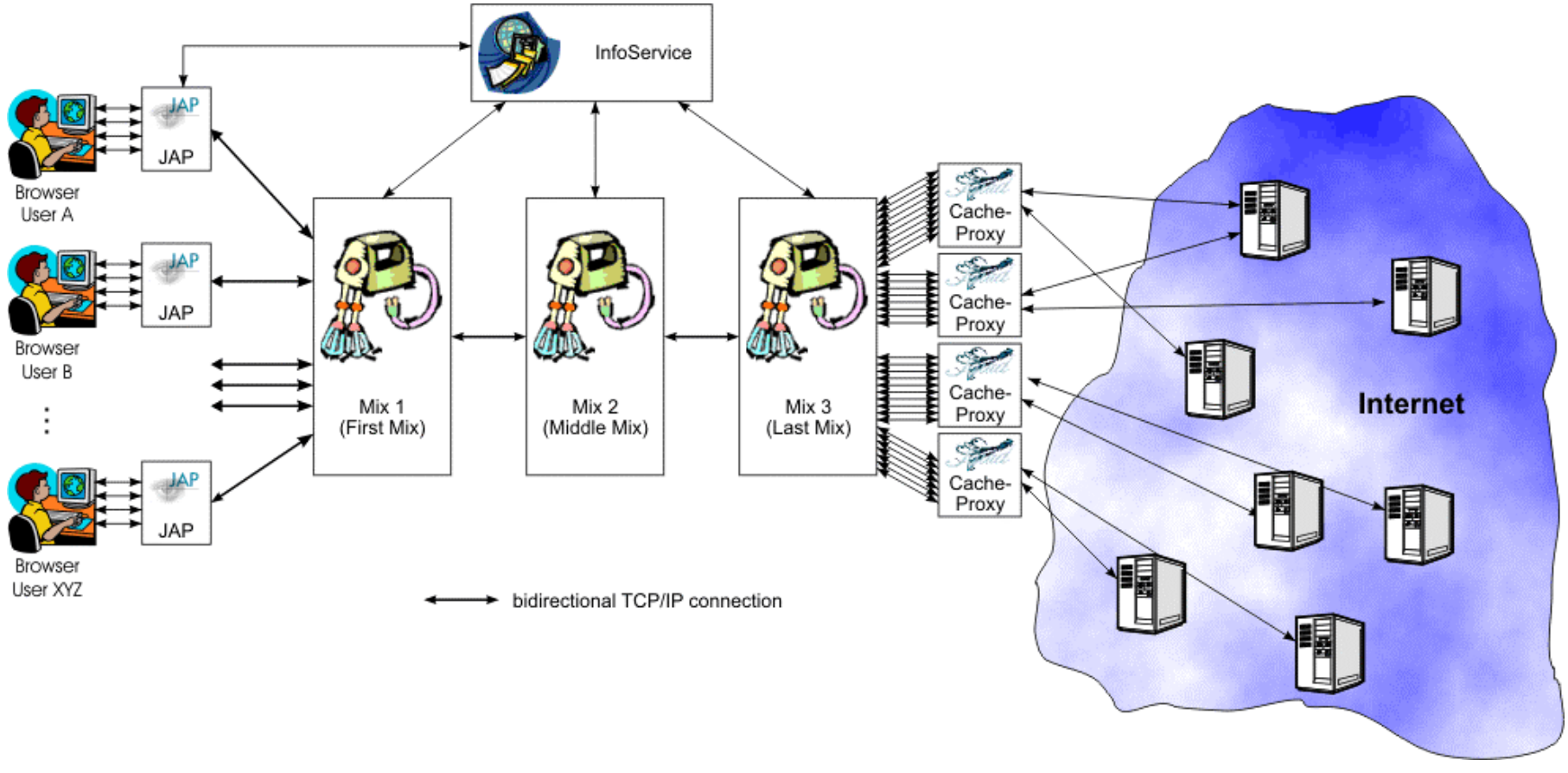
JAP – Web-Mixes

Stefan Köpsell, TU Dresden, sk13@inf.tu-dresden.de

- Overview
- Statistics
- Mix Development and Deployment
- How to attract Developers ?
- Attracting Users
- Abuse
- Results of a users' Survey

Overview


InfoService
JAP
JAP
Browser
User A
JAP
JAP
Browser
User B
JAP
JAP
Browser
User XYZ
Mix 1
(First Mix)
Mix 2
(Middle Mix)
Mix 3
(Last Mix)
Cache-
Proxy
Cache-
Proxy
Cache-
Proxy
Cache-
Proxy
Internet
bidirectional TCP/IP connection

Statistics

- open for public use since autumn 2000
- 1,3 Mio visits of our Web-Page http://anon.inf.tu-dresden.de
- > 200,000 downloads of JAP:
  - Windows : ca. 75 %
  - MacOS : ca. 3 %
  - Other : ca. 22 % [Linux, OS/2, Irix, Solaris etc.]
- 1,500–2,000 users concurrently online, maybe >30,000 in total
- 100 GByte traffic per day / 3 TByte traffic per month
- 10 Mio. URLs processed per day:
  - HTTP: >99,9% of requests >90% of traffic
  - FTP : < 0,1% of requests 5-10% of traffic
  - Targets: ca. 50% ".com" ca. 25% ".de" ca. 10% ".net" ca. 2% ".org"

Compared to other anonymous communication systems:

Is this little or much ???

Statistics

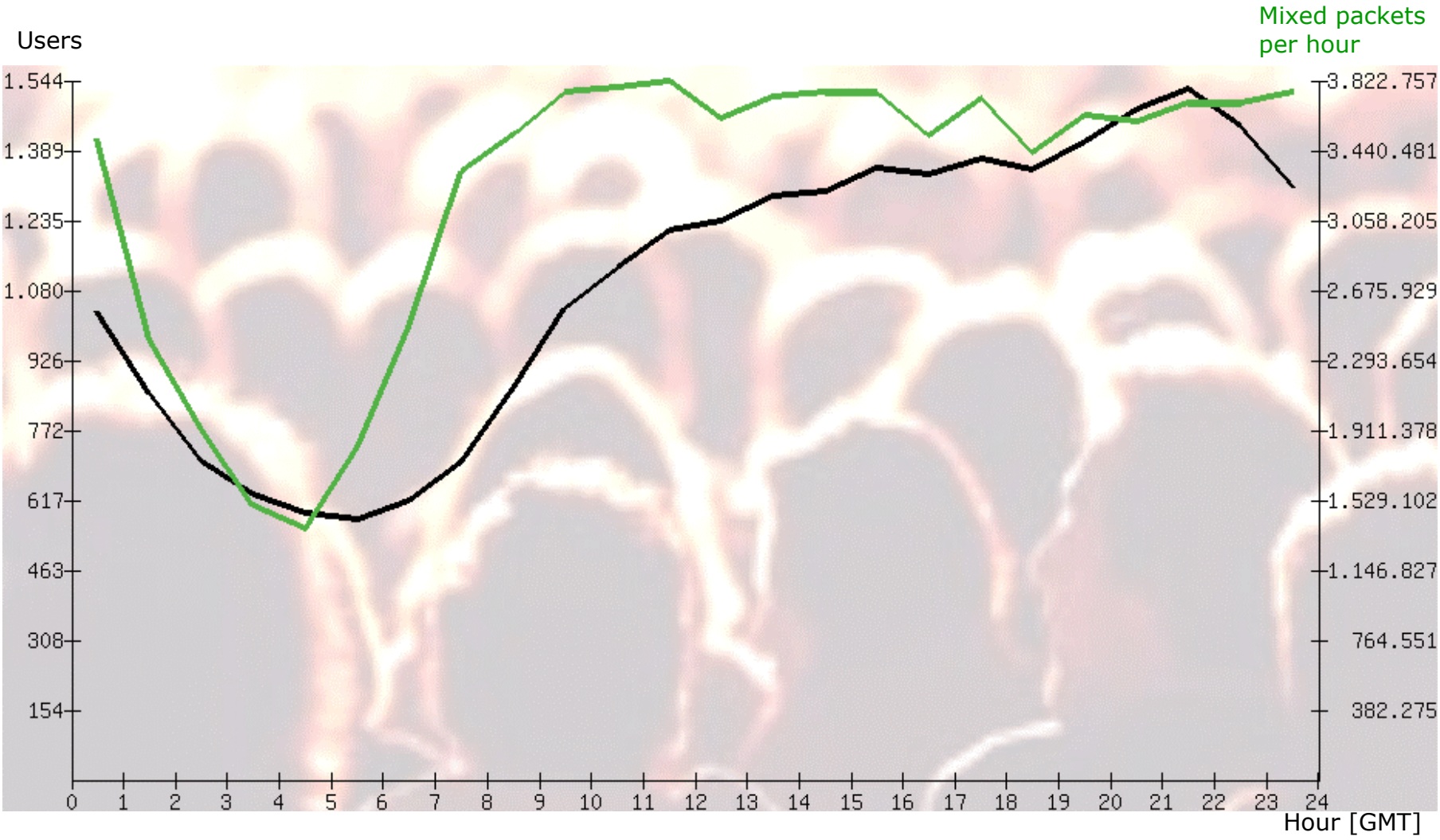
- open for public use since autumn 2000
- 1,3 Mio visits of our Web-Page http://anon.inf.tu-dresden.de
- > 200,000 downloads of JAP:
  - Windows : ca. 75 %
  - MacOS : ca. 3 %
  - Other : ca. 22 % [Linux, OS/2, Irix, Solaris etc.]
- 1,500–2,000 users concurrently online, maybe >30,000 in total
- 100 GByte traffic per day / 3 TByte traffic per month
- 10 Mio. URLs processed per day:
  - HTTP: >99,9% of requests >90% of traffic
  - FTP : < 0,1% of requests 5-10% of traffic
  - Targets: ca. 50% ".com" ca. 25% ".de" ca. 10% ".net" ca. 2% ".org"

Compared to other anonymous communication systems:

Is this little or much ???

Average usage

- Users and mixed packets over the day



Statistics

- open for public use since autumn 2000
- 1,3 Mio visits of our Web-Page http://anon.inf.tu-dresden.de
- > 200,000 downloads of JAP:
  - Windows : ca. 75 %
  - MacOS : ca. 3 %
  - Other : ca. 22 % [Linux, OS/2, Irix, Solaris etc.]
- 1,500–2,000 users concurrently online
- 100 GByte traffic per day / 3 TByte traffic per month
- 10 Mio. URLs processed per day:
  - HTTP: >99,9% of requests >90% of traffic
  - FTP : < 0,1% of requests 5-10% of traffic
  - Targets: ca. 50% ".com" ca. 25% ".de" ca. 10% ".net" ca. 2% ".org"

Compared to other anonymous communication systems :

Is this little or much ???

Mix Deployment

- 1. Approach
  - Assumption:
    - Mix operators are experienced system (unix) administrators
  - Conclusion:
    - Mix software installation and configuration need not to be easy
  - Results:
    - 1. Mix software is a command line program with many options
    - 2. Mix software comes as source code
    - ⇒ **The people who were willing to operate a mix failed.**

- 2. Approach
  - Assumption:
    - NOT all Mix operators are experienced system administrators
  - Conclusion
    - Mix installation and configuration hast to be as easy as possible

Mix Deployment

- Results:
  - Graphical user interface for Mix configuration written in Java (executable either as application or applet within your favourite browser)
  - Mix software is still a command line tool, but has only one option: the configuration file
  - Mix software runs on many platforms, so the operator can choose her or his favourite one
  - Try to use only components, which are included in the default installation of that operating system

- A new problem:
  - Configuration file is XML ⇒ we use Apaches Xerces-C++ XML-Library
  - Problems:
    - C++ ABI changed with every Version of GNU GCC, so precompiled versions of Xerces-C++ are often not usable
    - Changes in the Xerces-API (including namespace etc.) make it difficult to hold the Mix software compatible with all versions of Xerces

    ⇒ If people fail to compile the Mix the reason is Xerces!

    ⇒ Potential solution: Use other XML-Library like libxml, which is written in C

    … but this makes development more difficult

Easy development Û Easy deployment ??

Mix Deployment

- Results:
  - Graphical user interface for Mix configuration written in Java (executable either as application or applet within your favourite browser)
  - Mix software is still a command line tool, but has only one option: the configuration file
  - Mix software runs on many platforms, so the operator can choose her or his favourite one
  - Try to use only components, which are included in the default installation of that operating system

- A new problem:
  - Configuration file is XML ⇒ we use Apaches Xerces-C++ XML-Library
  - Problems:
    - C++ ABI changed with every Version of GNU GCC, so precompiled versions of Xerces-C++ are often not usable
    - Changes in the Xerces-API (including namespace etc.) make it difficult to hold the Mix software compatible with all versions of Xerces

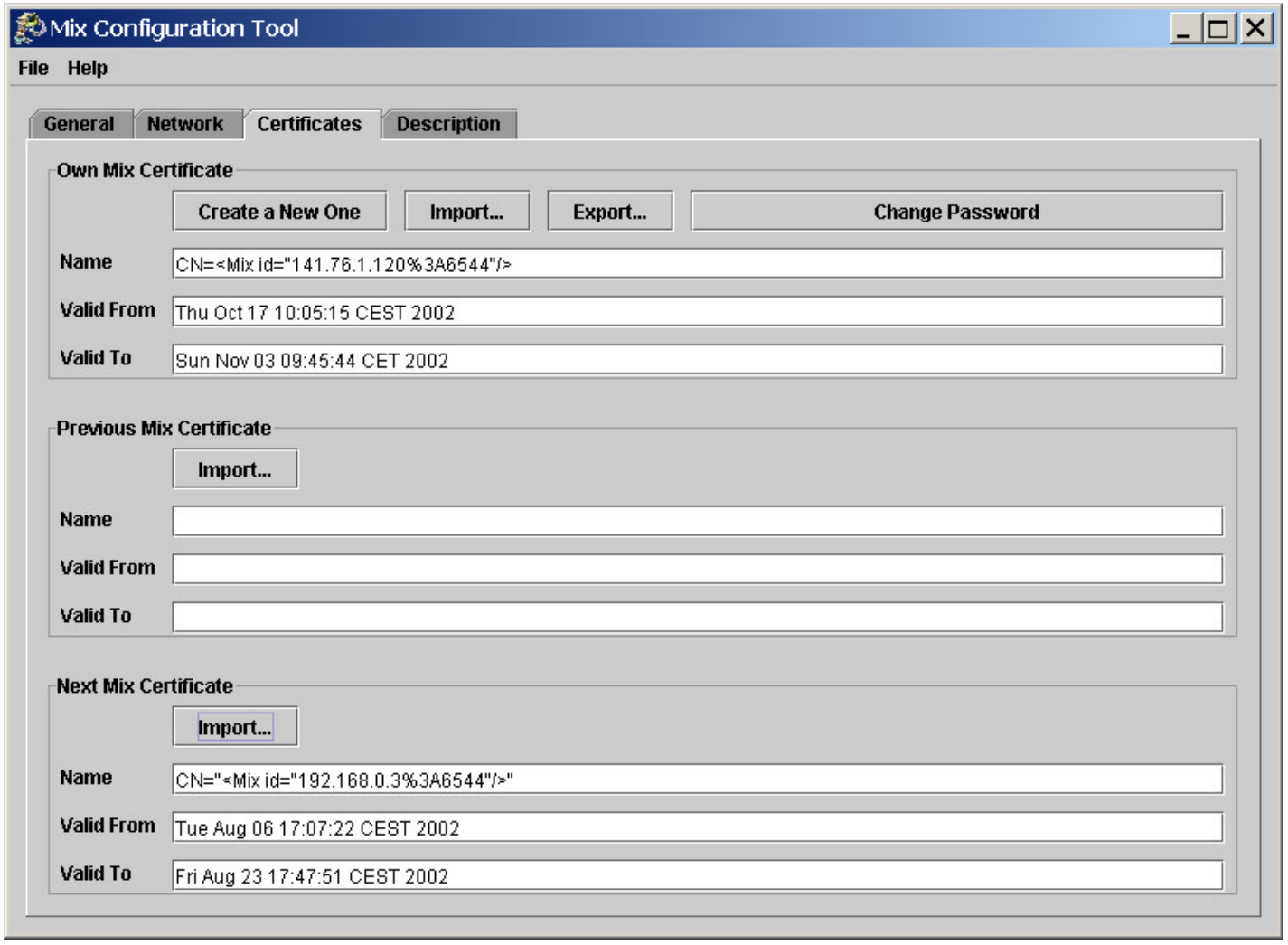
    ⇒ If people fail to compile the Mix the reason is Xerces!

    ⇒ Potential solution: Use other XML-Library like libxml, which is written in C

    … but this makes development more difficult

Easy development Û Easy deployment ??

Mix-Configuration Tool



Mix Deployment

- Results:
  - Graphical user interface for Mix configuration written in Java (executable either as Application or Applet within your favourite browser)
  - Mix software is still a command line tool, but has only one option: the configuration file
  - Mix software runs on many platforms, so the operator can choose her or his favourite one
  - Try to use only components, which are included in the default installation of that operating system

- A new problem:
  - Configuration file is XML ⇒ we use Apaches Xerces-C++ XML-Library
  - Problems:
    - C++ ABI changed with every Version of GNU GCC, so precompiled versions of Xerces-C++ are often not useable
    - Changes in the Xerces-API (including namespace etc.) make it difficult to hold the Mix-Software compatible with all versions of Xerces

    ⇒ If people fail to compile the Mix the reason is Xerces!

    ⇒ Potential solution: Use other XML-Library like libxml, which is written in C

    … but this makes development more difficult

Easy development Û Easy deployment ??

How to Attract Developers ?

- Coding the whole system (Mixes, JAP, InfoService etc.) needs really much resources (manpower)
- Idea: Using the power of the open source community to help
  - Whole project is open source (BSD style licence) and available at sourceforge.net
  - But: Attracting developers is not that easy (maybe because of the special research character of the project ?)
  - **How to attract developers ??**
  - **How is the development of other anon systems organized ??**

Attracting Users

- Support as many platforms as possible:
  - JAP is written in Java 1.1 and available for nearly every platform
  - Problems:
    - Java grants no access to system specific functions and configuration, e.g. changing the browser settings to use JAP as proxy is not possible
    - Real integration in the look and feel of a system is not possible
    - "write once, run anywhere" does not really work
    - **Solutions ??**
- Installation and configuration have to be easy:
  - If the user is not able to get it run within 10 minutes he will not use it at all
  - Most users like a graphical interface not a command line tool
- Give them support:
  - We have answered more than 5000 e-mails from users
    - **Has anyone experiences with tools supporting this ??**
  - Users are not willing to read anything like documentation, FAQs etc.
    - **How to force them reading before asking ??**

Attracting Users

- Firewalls are always a problem:
  - in companies "normal" users have no influence on the firewall configuration
  - Home users have many different kinds of personal firewalls and often do not know how to change their configuration
  - Our solution:
    - use only few connections to the outside world
    - design them in a way, that they could be tunnelled via common proxy protocols like HTTP, SOCKS etc.
    - let servers listen on usually "accessible" ports (80, 443 etc.)

    Other solutions ??
- We have made no "active" advertisement, but others report about the project on different media:
  - Newspapers, radio, TV, Internet etc.
  - Especially we get a push after each message on the German internet news board called "Heise News Ticker"
  - But: We believe, that at the moment most of our users are Germans, so

    What are the relevant media (especially internet based) for other countries ?
- We have exhibited on fairs like CeBIT
  - Although this also attracts users, using internet based media is much cheaper and results in more attention

Attracting Users

- "Hidden" functionality
  - People in countries with restrictive Internet access use the system just to freely browse the whole Web
  - Some countries have blocked our anon service
  - Big challenge:

How to make blocking as difficult as possible ?

- Keeping the system "alive"
  - Development and operating of the system cause great running costs
  - At the moment covered by the research project
  - But: How to recoup the costs afterwards ?
  - Are the users willing to pay, how much ?

Which experiences did commercial systems make?

Abuse

- Misuse of our anon service:
  - credit card fraud
  - blaming of people in postings to Newsgroups or Internet forums
  - identity theft
  - hacking of servers which run unpatched Microsoft IIS
  - 2-3 request per month from the police or public prosecutors
  - on request of site operators, we block them
  - Which experience did other anon systems make?
  - Should there be the possibility to reveal identities in certain situations (maybe according to the fairness assumptions of digital cash (e-coins)) ?
  - How to achieve this without monitoring all users?
  - In the sense of fairness, should the requested server be informed, that a certain request is anonymized (maybe by including a X-Anonymized header line) ?
  - Could this solve some abuse problems ?

- Abuse in Peer-To-Peer based systems:
  - in our system, only we get into contact with the police, but NOT our users (because the IP of the last node belongs to us)
  - this is different in Peer-To-Peer based systems like Crowds or Tarzan, because every participating user may be a “last node”
  - Is this a big problem for the acceptance of Peer-To-Peer based systems ?
  - Perhaps users would not risk to be contacted by the police ?

Results of a users' Survey

- Web based users' survey
- 4190 Entries from 07/04/2001 – 03/22/2003
- Results: (multiple choices are possible)
  - Reasons for using JAP:
    - 64% protection against the ISP
    - 51% protection against the police, secret service etc.
    - 47% protection against the operators of the Anon-Service
    - 34% free speech
    - 44% easy to use
    - 12% bypass censorship
  - 55% of the Users are willing to pay for JAP
  - 7% of the Users use JAP relating to business

Has anyone else made a survey relating to anonymous communication systems – and what are the results ??