MoSAIC –*http://www.laas.fr/mosaic/*

Kick-off Meeting, October 13th 2004

1. Introduction

This document summarizes the scientific and technical discussion that took place during MoSAIC's *kick-off* meeting on October 13th, 2004 at LAAS-CNRS in Toulouse. Attendees were:

- Yves Roudier from Eurecom;
- Michel Banâtre and Paul Couderc from IRISA;
- Laurent Blain, Ludovic Courtès, Yves Deswarte, Marc-Olivier Killijian, David Powell, Matthieu Roy and Isabelle Silvain from LAAS-CNRS.

The following sections describe short-term research tracks for the project and various ideas that popped up during the brainstorming session. These ideas are transcribed here in a *raw* way. Some of them fall under several themes and hence contain pointers to the other places where they appear.

In the following we use the terms *data owner* (respectively *data saver*) and *backup client* (resp. *backup server*) interchangeably.

2. Distributed Backup

MoSAIC will have to build on current archival, backup, and especially cooperative backup techniques. The following ideas were brought up and are to be further investigated (the first items may actually be considered as preliminary work):

1. **Typical use scenarios** need to be defined so as to determine a spectrum of realistic *internet connectivity* and *mobility* schemes; see also item 1 of section 3, p. 2.

2. **An interaction model between peers** should be defined based on a set of scenarios (e.g. frequency and duration of internet accesses? density of ad-hoc nodes in the absence of an internet access? usefulness of ad-hoc routing protocols and reachability of distant nodes?).

3. **Representing use cases and peer interactions as a finite state machine (FSM)** is a necessary first step to get an idea of how the overall system could work (e.g. Alice's PDA meets a peer PDA, introduces itself, negotiates the storage of 1k block for one week, etc.) and it could even provide a more detailed representation of the interactions between peers (see also item 8 of section 3, p. 2).

4. **Peer-to-peer file sharing and backup systems** are of particular interest as they provide approaches to *data dissemination, retrieval of scattered data* and *privacy and anonymity enforcement techniques* (among other things); peer-to-peer systems also implement various *economic models* that aim to give peers incentives to collaborate with the community.

5. **Fragmentation-Redundancy-Dissemination (FRD)** techniques are to be considered with respect to several scenarios; besides being crucial in enforcing data privacy (see item 2 of section 4, p. 3),

fragmentation is well suited for short peer interactions with frequent disconnections (one may not be able to send a whole file to a given peer); redundancy and dissemination help increase the data availability;

6. **Data backup operations** such as *incremental backups*, *evaluation of the differences between two version of a file*, *block-level backup* needs to be reviewed and compared to a data-semantic-aware backup technique and the advantages of *content-based block addressing*; see also item 10 of section 4, p. 3.

7. **Data recovery techniques** are to be considered in the framework of a "*push*" model (i.e. data saver pushes backed up data to its client data owners) versus a "*pull*" model (i.e. data owners query data savers as in peer-to-peer file sharing systems); see also item 9 of section 4, p. 3.

8. **Data revision and obsolescence management** raises a number of questions: dealing with the fact that only partial backups may be done while the data keeps changing; making sure that chunks coming from different revisions of a given file cannot be merged together when restoring the file; taking care of backing up entirely a given file revision before starting to back up chunks of later revisions; defining a mechanism allowing servers to determine which chunks of data are obsolete and can be deleted; a look at revision control systems such as CVS and GNU Arch would seem relevant here.

9. **Resource discovery** protocols may be used by a data owner to find out available storage space in its surrounding peers.

10. **Data transfer between mobile nodes** and in particular protocols addressing *ephemeral connection issues* (bootstrapping, abortion, etc.).

11. **Data ownership and sharing** is a major concern: while some data are definitely private and should not be made available to other peers, the backup system could benefit from knowing that *some* data were created *collectively* (e.g. during a meeting using collaborative applications); moreover such collectively created data should be made available to all the peers which participated in their creation; this may require users to explicitly attach *semantic information* to their data; see also item 6 of section 4, p. 3.

12. **Data criticality levels** may be used to provide the backup system with hints on which data should be backed up first; again, the user would have to explicitly provide the system with this information; managing criticality levels may require a "smart backup scheduling" technique: if the most critical data are stored in very active files it may turn out that the system keeps backing up the different versions of those files while completely forgetting about less critical data.

13. **Useful online services** may improve the backup functionality (e.g. whenever a data owner can access its trusted desktop computer, it might want to send its most critical data there; similarly, data savers might *push* data back to the data owner's trusted desktop computer's whenever the latter becomes available); see also item 7 of section 3, p. 2.

3. Negotiation and Collaboration

This section focuses on ideas related to mechanisms (economic models) that aim to provide peers with *collaboration incentives* while protecting the system against *selfishness "attacks"*.

1. **Typical use scenarios** will be helpful in finding out *how* peers could interact with each other; see also item 1 of section 2, p. 1.

2. **Tamper-proof hardware and smartcards** may be used as a means of providing a *secure and reliable user identification* which might help enforce data privacy (see item 1 of section 4, p. 3) and may help implement *electronic-cash based incentives* (which require the human owner – rather than his device – to be reliably identified and reliance on the impossibility to forge identities).

3. **Identifying devices and/or users** is needed in order to implement an economic model; see also item 5 of section 4, p. 3.

4. **Identification through a trusted certification authority** may be used; however, using a trusted authority breaks the peer-to-peer model and may be impossible for disconnected mobile nodes which do not have an internet access.

5. **A hybrid model based on ecash *and* a trust-based economic model** may be considered given that mobile nodes may not always be able to connect to trusted third-parties such as the "bank" that issues cash and validates transactions.

6. **Trust establishment** is an issue as nodes have no prior mutual trust (if a central certification authority is to be involved in the process, then peers may want to ask it to validate a peer's identity as soon as they can; if a trust-based decentralized approach is considered, trust has to be bootstrapped somehow so that peers can start collaborating together); see also item 7 of section 4, p. 3.

7. **Useful online services** that may help the backup service (e.g. trusted third parties such as a bank or a certification authority) need to be identified; see also item 13 of section 2, p. 1.

8. **Representing use cases and peer interactions as a finite state machine**, as mentioned in item 3 of section 2, p. 1, may help describe peer interactions at various levels of abstraction.

9. **Cooperation, negotiation**: peers will have to negotiate storage space and possibly duration; see also item 8 of section 4, p. 3.

10. **ID bootstrapping and recovery**: in order to start using the system, users may have to acquire a unique ID from a central authority or devices may identify themselves using a vendor-defined identifier (e.g. MAC address); after loss, theft or crash of a mobile device, it may desirable for the user/device to reenter the system and automatically benefit from the trust peers had in him/it previously and/or use the electronic cash he/it had previously accumulated; see also item 11 of section 4, p. 3.

11. **Quality of Service**, in particular making it possible to guarantee that data will be kept for some time and that its *criticality level* (see above) can be accounted for.

4. Privacy

This section deals with techniques useful for guaranteeing backed up data privacy. Some of the ideas presented in previous sections are relevant to this goal, most notable data fragmentation and dissemination, data ownership and sharing management, as well as user/device identification. Below is a list of more specific ideas.

1. **Tamper-proof hardware and smartcards** may be used to enforce data privacy (e.g. the user's private key is stored on a smartcard which can be reused after a device failure, therefore making it possible for the user to retrieve its previously backed up data); see also item 2 of section 3, p. 2.

2. **Data fragmentation, dissemination and encryption** mentioned earlier (item 5 of section 2, p. 1) is a means of ensuring data privacy since no single data saver has enough information to reconstruct a backed-up file, nor is it practically feasible for a malicious data saver to decipher data blocks.

3. **Allowing users to have serveral IDs or roles** might help in providing *anonymity*; however, even though a client system could use several application-level identities when connecting to a backup server, the server may still be able to retrieve the client's actual MAC (hardware) address or some such; anonymity seems a lesser requirement in a backup system than in a document publishing system.

4. **Security policies** may define who or which peers can access certain data; this is similar to the data ownership issue mentioned earlier.

5. **Identifying devices and/or users** may be a crucial point in order to make private data only available to authorized users (provided snapshots are encrypted and disseminated, identifying *snapshots* rather than devices and/or users might be sufficient since access control can be left to the user who can decide whether to disclose a snapshot ID); see also item 3 of section 3, p. 2.

6. **Data ownership and sharing** semantics must be defined in order to guarantee data privacy (see also item 11 of section 2, p. 1).

7. **Trust establishment** is an issue wrt. privacy: it should be impossible to *forge* new identities or to use another device/user's identity; see also item 6 of section 3, p. 2.

8. **Cooperation and negotiation** should not reveal private data; see also item 9 of section 3, p. 2.

9. **Data recovery techniques** are a crucial point for data privacy enforcement: it should be practically infeasible for someone (including backup server owners) to retrieve and decrypt backed up data, unless entitled to do so; see also item 7 of section 2, p. 1.

10. **Data backup operations** should be made *opaque* (using ciphering techniques) so that external observers may not be able to know what data are being backed up; see also item 6 of section 2, p. 1.

11. **ID bootstrapping and recovery** is a potential issue wrt. privacy: it should be impossible for someone to identify herself/himself as another person so as to make backup operations on his/her behalf (thus potentially altering the trust other people have in him/her) or to retrieve her data; see also item 10 of section 3, p. 2.

12. **Worms such as those developed at Xerox PARC in the early 80's** might have had similar concerns to those that we have now.

5. Experimentation

Design proposals and actual prototypes shall be evaluated keeping the following items in mind:

1. **Performance and efficiency metrics** need to be defined in order to evaluate the backup system; we may as well define a set of *benchmarking scenarios*.

2. **Validation using proofs** is a prerequisite for confidence in a given design, protocol and interaction model.

3. **Safety and liveness properties** for the backup system must be defined (e.g. under what circumstances should we expect the system to be able/unable to retrieve backed up data; etc.).

4. **Technology issues** such as choosing the most convenient software platform for building prototypes have to be solved (e.g. whether using a Java environment induces too many limitations on what can be done, whether actual Java implementations are available for the chosen OS and architecture, etc.).