somehow. It was also felt that the heavily polymorphic (thus pointer-based) APIs in use for calendaring in particular would make it very hard to serialize the data, which would be needed for the transfer over DCOP. Transferring the data through an IPC interface all the time might also end up being slow, and this was raised as a concern as well, especially if the server was also going to be used for access to email, which had been suggested as an option.

The conclusion of the meeting was that the memory footprint problem might be better solved through a more clever sharing mechanism for both the on-disk cache and the in-memory representation, possibly through the use of memory mapped files. The complexity of changing to a client/server-based architecture was deemed too challenging, and it was felt that the benefits that would come from it were not enough to justify the risk of breaking an essentially working (albeit not satisfactorily) system. As a possible alternative, the Evolution Data Server (EDS) as used by the Evolution team, a competing PIM suite associated with the GNOME project and written in C using the glib and GTK library stack, was agreed to be worth investigating.

The proponents of the data server idea left the meeting somewhat disappointed, and in the following months not much progress was made one way or the other. A short foray into EDS's codebase with the aim of adding support for the libraries used by KDE for the address book ended quickly and abruptly when those involved realized that bridging the C and C++ worlds, and especially mindsets and API styles, to arrive at a working solution would be inelegant at best and unreliable and incomplete at worst. EDS's use of CORBA, a technology initially adopted by KDE and then replaced with DCOP for a variety of reasons, was not very appealing either. It should be noted, in all fairness, that the rejection of EDS as the basis for KDE's new PIM data infrastructure was based on technical judgement as well as personal bias against C, a dislike of the implementation and its perceived maintainability, along with a certain amount of "not invented here" syndrome.

Toward the end of 2005, the problems discussed at the beginning of the year had only become more pressing. Email was not easily accessibly to applications such as desktop search agents or to semantic tagging and linking frameworks. The mail handling application, including its user interface, had to be started in order to access an attachment in a message found in a search index and to open it for editing. These and similar usability and performance issues were adding to the unhappiness with the existing infrastructure.

At a conference in Bangalore, India, where a significant part of the team of developers working on Evolution was located at the time, we had the opportunity to discuss some of these issues with them and ask them about their experiences with EDS. In these meetings it became quickly apparent that they were facing many of the same issues the KDEPIM team had identified, that they were considering similar solutions, and that they did not feel that extending EDS to support mail or porting it away from CORBA were feasible options. The general message from the Evolution team was that if the KDEPIM developers were to build a new infrastructure for PIM data access, they would be interested in sharing it at least in concept, and possibly also in implementation, provided that it wasn't available solely as a C++ and KDE library.