

itself. The main technical goal, to support the developers and other contributors with funds, materials, organizations, and other resources, are comparatively simple to achieve.

The result is a living, active, vibrant community with an established development process, a stable and healthy fluctuation of contributors, and a lot of fun on the way. And this in turn helps to attract and secure the most important resource for such an environment: new contributors. KDE has succeeded in preventing the typical tendency to create larger and larger formal structures and positions and also does not have any dictators, as benevolent as they might be. KDE is an archetypal Free Software community.

In order to understand how this community functions in practice, how architectural decisions are arrived at, and how the peculiar process influences the outcomes technologically, we will look at two examples in some detail: Akonadi, the personal information management infrastructure layer for KDE 4, and ThreadWeaver, a small library for high-level concurrency management.

## Akonadi

The KDE 4 platform, both as a development platform and as a runtime environment for the execution of integrated applications, rests on a number of so-called “pillars.” These key pieces of infrastructure provide the central services that applications expect to have easy and ubiquitous access to on a modern desktop. There is the *Solid* hardware interaction layer, which is responsible for providing the desktop with information about and notification from the hardware, such as a USB stick becoming available or the network going down. *Phonon* provides an easy-to-program multimedia layer for playback of various media types and user interface elements for their control. *Plasma* is a library for rich, dynamic, scalable (SVG-based) user interfaces that go beyond the standard office look.

The personal information of the user—her email, appointments, tasks, journal entries, blog posts and feeds, bookmarks, chat transcripts, address book entries, etc.—contains not only a large amount of primary information (the content of the data). It also weaves a rich contextual fabric from which much about the user’s preferences, social interactions, and work contexts can be learned, and it can be used to make the interaction experience of many applications on the desktop more valuable and interesting, provided that it is readily, pervasively, and reliably accessible. The *Akonadi* framework aims to provide access to the user’s personal information, the associated metadata, and the relationships between all that data, as well as services that operate on them. It aggregates information from a variety of sources, such as email and groupware servers, web and grid services, or local applications that feed into it, caches that information, and provides access to it. Akonadi thus serves as another pillar of the KDE 4 desktop but, as we shall see, aims to go beyond that.

In the following sections, we will explore the history of this large and powerful framework, the social, technical, and organizational struggles that were and are involved in making it