processor vendors that Xen can run unmodified operating systems, such as Microsoft Windows. One of the new frontiers in processor development is the addition of new features to support virtual machines and remove some of the performance penalties.

The architecture of Xen is slowly evolving as new features are added and new hardware becomes available. However, the same basic structure has persisted from the original prototype through to the present version. In this chapter, we trace how Xen's architecture has matured from its early days as a research project, through three major versions, and to the present day.

Xenoservers

Work on Xen began at the University of Cambridge in April 2002. It was initially developed as part of the Xenoservers project, which aimed to create a "global distributed computing infrastructure."

Around the same time, *grid computing* was being advanced as the best way to make use of computing resources that are scattered throughout the world. The original grid proposal cast computer time as a utility, like electricity, which could be obtained from a grid—or network—of collaborating computers. However, subsequent implementations concentrated on *virtual organizations*: groups of companies and institutions that established possibly complicated relationships of trust, which are enforced by heavyweight public-key cryptography for authentication and authorization.

Xenoservers approached the problem from the opposite direction. Instead of forging trust relationships with service providers, the customer chooses a resource on the open market through a broker known as a *XenoCorp*. The XenoCorp stores a list of *xenoservers*—computers offered for lease by third parties—and matches customers with servers, collecting and passing on payment for the utility. Crucially, there is *mutual distrust* between the customer and the provider: the customer cannot harm the provider's machine, and the provider cannot tamper with the customer's job.

TRUST

It might sound counterintuitive that *distrust* is a useful architectural feature. However, the main goal of security in this context is to prevent other individuals from accessing or interfering with your sensitive data. A *trusted* system, then, is one that is allowed access to your data. When distrust is built into the architecture, the number of trusted components is minimized, and this therefore provides security by default.