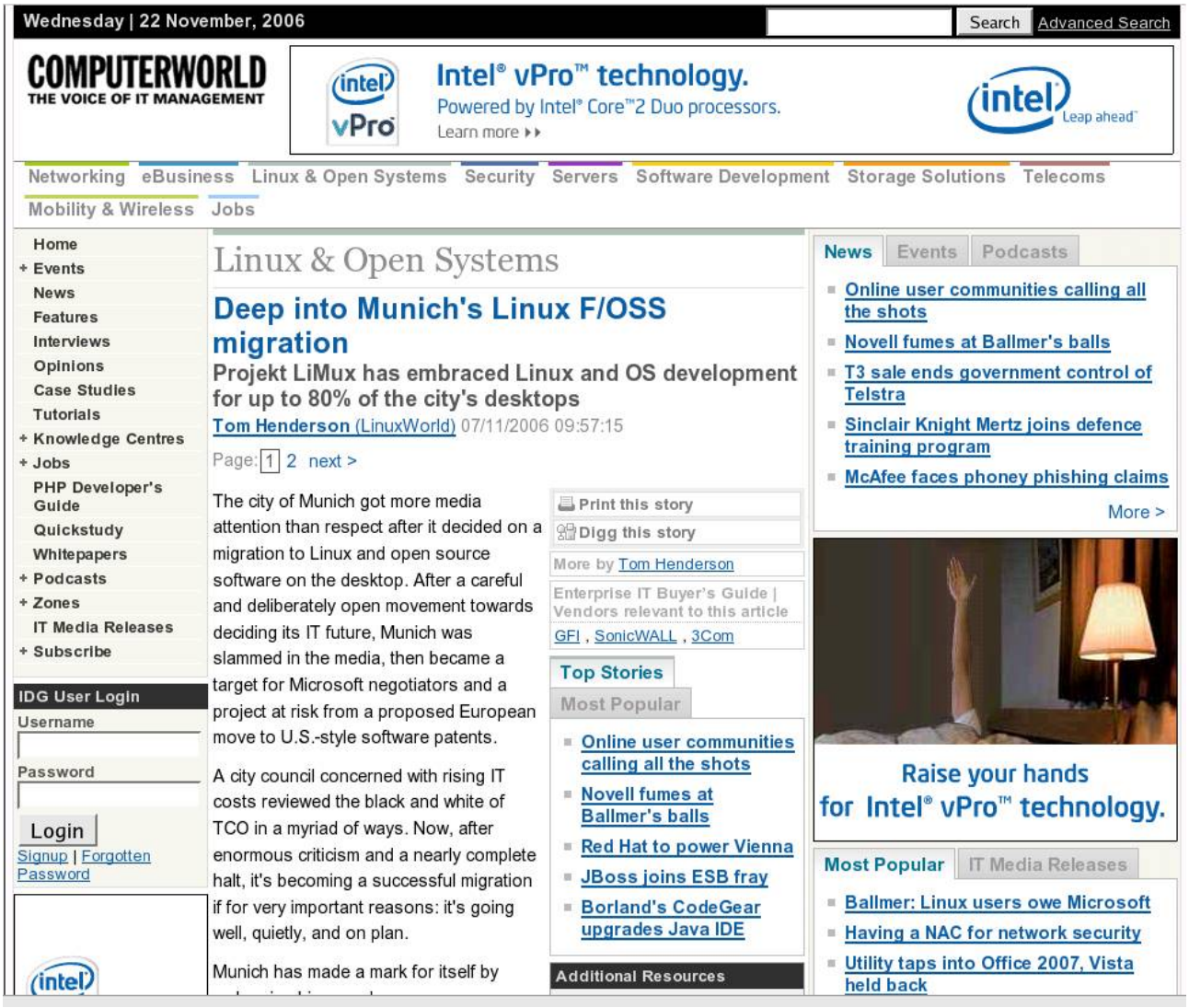

The city of Munich got more media attention than respect after it decided on a migration to Linux and open source software on the desktop. After a careful and deliberately open movement towards deciding its IT future, Munich was slammed in the media, then became a target for Microsoft negotiators and a project at risk from a proposed European move to U.S.-style software patents.

A city council concerned with rising IT costs reviewed the black and white of TCO in a myriad of ways. Now, after enormous criticism and a nearly complete halt, it's becoming a successful migration if for very important reasons: it's going well, quietly, and on plan.

Munich has made a mark for itself by embracing Linux and open source development for up to 80% of the city's 16,000 desktops. The remaining user desktops will use Microsoft Windows XP for apps that have no open source equivalent, including AutoCad). Mac OS currently has no future in the city's IT plans. Although seemingly controversial, city officials claim that the arduous process that led to the decision to migrate to Linux was actually based on Microsoft's policies on Windows NT, and a subsequent study to determine the best course of action pursuant to the unexpectedly short life support cycle for NT.

Microsoft announced an end-of-life support plan for NT that would prevent the operating system from surviving through the life cycle than the IT officials in Munich had anticipated. Faced with new, unanticipated licensing costs, the Munich City Council then commissioned a study of returns based on operating systems and applications product cost, life cycle of deployment, and usability for user desktops. The city already ran server applications on Unix, on a Siemens mainframe and Sun equipment, and many of these might need to be changed, too.

The decision could have swayed towards Microsoft infrastructure -- operating system and Office apps -- if the view were taken on a short-term deployment life cycle, said Florian SchieBl of Munich's Projekt LiMux. When the long-term benefits were weighed in, open source was more cost-effective. That's what swayed Munich's city council: a medium- and long-term value and TCO assessment.

The results were tested both critically and vociferously. Political forces, combined with pressure from Microsoft, created a tense atmosphere. An initial contract formulated the migration steps and information that would help construct policy and specifics of LiMux recipes and use. Users were familiar with Microsoft, NT and Office. If a migration were to take place, the knowledge capital, not to mention existing assets in terms of procedures and documents would need parity with a subsequent platform. Documents would need conversion, procedures for exchanging data with Microsoft platforms would change, and all of the chasms of migration would need to be crossed to make the project work.

Munich immediately took a lot of criticism for its choices, but also joined a number of smaller European governmental authorities that either came to similar conclusions, or started based on other values.

Inside Projekt LiMux

Once the F/OSS-Linux decision was made, the city drew up numerous requirements and began a bid process for selection of a distribution. Various applications, Linux distributions and package combinations were considered until a suitable mix of applications and operating system components was found. After testing, these components would be put together for a trial, then an initial rollout would occur. IBM and SUSE/Novell initially helped guide the project's concepts in terms of application examination and migration tasks and issues. In the end, however, the base of LiMux has become Debian as the operating system core, coupled to KDE and OpenOffice.

As SchieBl explains, Munich needed a subset of what commercial distributions offer. Software patents were a great concern if the entire LiMux distribution would be subsequently freely distributed if others wanted to use it. This required examining all of the components for licensing and distribution rights. Clearing the software patent hurdle caused the project to grind to a halt during that phase, and delayed the process considerably. Although software patent issues still arise from the dead in the European Union political process, Munich feels it has cleared them sufficiently to deploy LiMux desktop systems.

The trial was tested and completed, and a first rollout occurred on Sept. 19, 2006, about three years after the decision to make the move towards Linux. The migration is scheduled to take place, subject to development of user applications and other factors, by approximately 2009. More than 170 applications will be replaced with open source.

LiMux components

Operating system: The core operating system is Debian GNU/Linux Sarge with the 2.6.17 kernel. It's a stripped version that doesn't use the often crammed-with-seldom-used-freebies distribution methodology employed by 'commercial' releases of Linux. Instead, a sparse, conservative approach is used. Munich is lucky to have Debian programmers and kernel hackers in the local area to call on for help and expertise. It's no coincidence that the last Debian release party was in their building, SchieBl says.

Office applications: Projekt LiMux uses OpenOffice.org 2.0.4 for 'office automation' applications. Interoperability with current Microsoft Office 97 documents is reasonably assured. Desktop KDE 3.5 and Xorg are the basis for a graphical user interface, not the Xfree86 originally included in Debian's

Sarge release. Mozilla Thunderbird 1.5.x and Firefox 1.5.x are used for mail and browsing.

Mail and X.500 authentication service is performed by a Critical Path mail server, and a replacement e-mail service may or may not take place in the future. The LiMux model doesn't use a thin-client approach focused towards heavy server-based applications, rather a mixture of client side application s coupled to Web and client-server applications. The 'fat client' approach pushes processing towards the selected client applicationss, rather than browser-based analogs of 'fat client' applications.

Desktop hardware: The current standard issue is a standard 32-bit i386 machine (64-bit in the future) with a 40G to 80GB hard drive, 512MB of dynamic RAM, and a 15 to 17-inch flat screen display. The type of machine varies, changing every six months by contract. USB drives (flash sticks/dongles) are permitted, and other user hardware add-ons are allowed and are departmentally approved.

Network profile and resources: Network use methodology is by fat client, except for PC access through APIs to Oracle (on Sun hardware) or Adabas (on a Seimens BS2000) mainframe via terminal or Web-based connectivity. Connectivity and authentication comes via CriticalPath's X.500 directory services and Sun's PC-Netlink with Lightweight Directory Access Protocol.

Munich uses an IPV4 network infrastructure with standard Cisco routers. Mobility is allowed, but not home-worker VPN connectivity. This simplifies security concerns at the possible risk of reduced productivity, some would say, but carrying home USB flash/memory drives is approved for data transfer where it's permitted by departmental policy and governing law.

Key to productive administration and system delivery is a new tool, built for Projekt LiMux. Debian's Fully Automatic Installation (FAI), a PHP script, is integrated into a tool developed by Gonicus tool called GOsa2. GOsa2 allows LiMux administrators to install preconfigured client payloads from an image repository, a great convenience. GOsa2 also uses a database for systems and application control settings for desktops, such as display resolutions, desktop-specific printing settings and installed applications.

Oddly, there's not a great deal of connectivity between the municipal network and the three major universities in Munich. Instead, Munich's hotbed of F/OSS developers have been lending time and expertise towards the success of Projekt LiMux largely outside of the auspices of the local universities.

LiMux intangibles

More than a third of the overall budget allocation in the LiMux project is dedicated towards training for LiMux, its applications, and towards online learning resources. Team leaders of Projekt LiMux have designed their own training methodology, done in modular/topical format. Personal training comes first, then eLearning/CBT/online reinforcement.

As an example, basics are covered for common OpenOffice matters such as file management (open, closing, searching, and so on), then specifics, such as building and managing forms, is a specific eLearning module.

LiMux futures

Munich faces the same issues that other adopters of free and open source software must deal with: the specter of constantly re-arising software patent issues, application conversion and migration issues, and the gradual weaning process from historically used applications.

There's an additional dependency on core application providers, both the fruits of the Debian community (known for its deliberative conservatism), the evolution of OpenOffice, as well as the

myriad other applications that are in use today, and might be envisioned for the future. Migration from existing applications, and the training processes needed to bring users to a comfort level for the mainstream choices made for Projekt LiMux is a huge part of both the budget and effort behind the migration in Munich, a step that may help to ensure its success.

Projeckt LiMux is a bold and brave move, but one that was also conceived in an open and highly deliberative process. It's for those reasons, and a listener's attitude that will help ensure its success.

Henderson is principal researcher and managing director of U.S.-based Extremelabs.