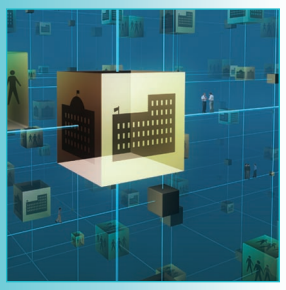


Virtual Organization Support within a Grid-Wide Operating System

Massimo Coppola
Consiglio Nazionale delle Ricerche (ISTI-CNR)

Yvon Jégou
Institut National de Recherche en Informatique et Automatique (INRIA)

Brian Matthews
Science and Technology Facilities Council

Christine Morin
Institut National de Recherche en Informatique et Automatique (INRIA)

Luis Pablo Prieto
Telefónica Investigación y Desarrollo

Óscar David Sánchez
Institut National de Recherche en Informatique et Automatique (INRIA)

Erica Y. Yang
Science and Technology Facilities Council

Haiyan Yu
Chinese Academy of Sciences

Despite grids' popularity, virtual organizations (VOs) have yet to become a commodity technology in modern computing environments due to the complexity of managing them and difficulty of assuring user and VO isolation. Here, the authors describe the VO management approach taken by XtreemOS, a new grid operating system with native support for VOs that supports a wide range of computing resources, from clusters to mobiles. They also discuss the requirements for the VO model and management within XtreemOS and introduce an expandable VO model and a system architecture that supports it.

Grid middleware has become an integral part of scientific computing. The notion of virtual organizations (VOs) is essential for computational grids with large numbers of users and computing nodes. A VO is a set of users and real organizations that collectively provide resources they want to exploit for a common goal. In grid computing, physical machines, services, applications, and data sets, just to name a few, can all be seen as resources. With regard to VOs, the ability to provide a composite platform on which users can arbitrarily run their applications is an ambitious common goal, although we shouldn't disregard supporting VOs with a short lifespan or very specific aim.

VOs achieve synergies by grouping users that share such an enlarged set of resources. Resource access and sharing in a typically multidomain and heterogeneous environment is no trivial task if we want to provide even basic authentication, security, and resource management functionalities. VO use has yet to become commercially widespread due to the complexity of managing VOs and the difficulty of ensuring mutual isolation among different users and VOs.

XtreemOS (www.xtreemos.eu) is an emerging European project with an aim to design, implement, evaluate, and distribute an open source OS that supports grid applications and runs on a range of platforms, from clusters to mobile devices. The goal is to provide an abstract interface to remote as well

1089-7801/08/$25.00 © 2008 IEEE

as local resources, the way a traditional OS does for a single computer. It's thus clear that VO support and management is central in XtreemOS and intertwined with the general system design and implementation. As researchers involved in developing the XtreemOS security and VO management functionalities, we will take a look at how VOs fit into XtreemOS and peek behind the mask to find out how the whole system can be implemented.

The XtreemOS Approach

XtreemOS is based on the existing Linux OS, and one of its key features is its support for VOs. However, the exact realization of a VO differs depending on its application: some approaches focus on the legal or contractual arrangements between participating entities, whereas other task-oriented approaches emphasize the workflow to achieve a goal. VOs can range from long-lived collaborations with several users (as in large-scale scientific applications) to short-lived, dynamic ventures among a few participants to achieve one task (such as in commercial scenarios). A general-purpose grid OS should take a flexible approach to support a wide range of applications. XtreemOS assumes a minimal definition of VOs and provides a toolbox that system administrators can configure to the needs of their users and applications.

In XtreemOS, a set of system services, extending those found in Linux, provide users with the capabilities associated with grid middleware. This native support means that XtreemOS will significantly ease the task of managing and using VOs without compromising efficiency, flexibility, or backward compatibility. Using this approach, administrators will find that setting up VOs is relatively simple because the tools are packaged into a single distribution with "one-click" configuration. Users don't need to learn new interfaces and tools to use VOs given that most tools provide them with standard Unix commands with which they're already familiar. Applications won't need to be refactored to run on VOs given that most XtreemOS APIs are POSIX (Portable Operating System Interface based on Unix)-compliant. Thus, many barriers to using VOs will be overcome.

Challenges

One of the main guides throughout the design and implementation of XtreemOS is the requirement analysis of 14 scientific and enterprise business applications[1] performed in an early phase of the XtreemOS project. From that, and after investigating current grid VO solutions, we derived several key challenges that a VO model and implementation should solve to be widely adopted.

Interoperability with other frameworks. Several different VO management frameworks and security models exist, and new ones continue to emerge. Their implementations vary in how they represent different user identities (such as X.509 certificates and Shibboleth handles [http://shibboleth.internet2.edu]), information exchange protocols (such as push, pull, and agent models), different representations of security attributes (such as proxy certificates [www.ietf.org/rfc/rfc3820.txt] or SAML tokens [www.oasis-open.org/committees/security]), and different access-control models (such as role-based access control).

In XtreemOS, VOs must interoperate with existing solutions and traditional system security mechanisms (such as Kerberos) rather than replace them.

Customizable isolation, access control, and auditing. A secure grid system must provide strict access control from the service level down to the system-object level. Aside from extending local access control to obey VO-mandated policies, our project has to provide the strong isolation properties required in many commercial applications.

Hiding user identities, protecting files and processes, and enforcing and hiding performance load are difficult to implement without OS support. Some of these functionalities bear an overhead that's acceptable only in certain contexts. In all cases, however, administrators should be able to monitor and log OS service usage, as well as system-object access, in a way that ensures nonrepudiation of logged information.

Scalability of dynamic VO management. In grid VO implementation, there are considerable scalability issues with respect to the system's performance and its ability to adapt to changes. Because a resource node might provide access to thousands of grid users from multiple VOs, the local OS must provide strong isolation

properties. When VOs are dynamically created or changed, maintaining consistency of static, local configurations becomes a complex task and a heavy administrative burden, even if partially automated.

Thus, to support large numbers of users in such a dynamic environment, XtreemOS avoids implementation solutions that rely on local configuration files, which statically contain user or resource information.

Ease of use and management. Typical authentication systems found in existing grid middleware are independent from security infrastructures found in most OSs.

With grid middleware such as Globus (www.globus.org) or Glite (http://glite.web.cern.ch/glite/), a user must explicitly acquire and manage at least two independent identities: a local identity that's significant for the OS initiating the grid request, and a grid identity to authenticate to the grid. These identities are usually issued by separate institutions and supported by different security technologies, such as Kerberos and Public Key Infrastructure (PKI). Manually managing multiple credentials can be a daunting task for nontechnical users, who often have limited experience with PKI.

Design Principles

In XtreemOS, support for VOs consists of several design principles that the XtreemOS team derived from the overall project guidelines and gathered requirements.

Single sign-on. In XtreemOS, we approach credential management by integrating the grid-level authentication with system-level authentication. Because XtreemOS is a grid OS, it provides users with single-sign-on (SSO) access to grid resources. From a security perspective, users don't need to be aware of the extra authentication steps, thus achieving an enhanced level of transparency to the grid.

Independence of user and resource management. In line with XtreemOS's fundamental approach, we developed a VO-centric security solution — a set of security services that keep user management cleanly separated from resource management within the VO scope.

Because of this separation, adding or removing VO users won't significantly affect performance and won't change configuration of resource management in a VO, and vice versa. Nevertheless, we take the advantage of integrating such a solution into system-level services.

Dynamic mapping between VO and Unix entities. To achieve maximum backward compatibility, XtreemOS adopts a systematic approach to extend Linux with VO support, which is almost transparent to upper-level services and applications. This requires a set of mechanisms to translate grid users to local Unix User and Group Identifier (UID/GID), to translate VO policies into Unix authorization rules, and to translate access-rights attributes into Unix access rights. The approach lets the users execute unmodified legacy applications on an XtreemOS computer, while taking advantage of the VO architecture.

Minimized changes to the Linux kernel. The XtreemOS project aims to minimize changes to the Linux kernel, thereby improving user and developer acceptance and simplifying the task of maintaining the XtreemOS code contribution as the mainstream Linux kernel evolution proceeds. Where kernel-level operation is needed, XtreemOS should achieve it by exploiting standard modular subsystems of the Linux OS.

XtreemOS VO Management

So far, the XtreemOS VO management (VOM) architecture hasn't actually required any change to the kernel code. VOM is composed of a set of security services running on top of Linux, and a set of lower-level mechanisms at the kernel level. Now let's take a look at the high-level, distributed architecture of XtreemOS VOM services, and then at the underlying machinery that, by exploiting the flexibility of specific Linux subsystems lets us configure a kernel instance to seamlessly perform the mapping between VO entities and Unix entities.

XtreemOS VOM Architecture

VOM provides a logical grouping of the infrastructural services needed to manage the entities involved in a VO and ensure a consistent and coherent exploitation of the resources, capabilities, and information inside it. VOM covers five services: identity, attribute, credential, membership, and policy. Given that these services support XtreemOS's authentication and

authorization infrastructure, we refer to them as security services.

Figure 1 illustrates XtreemOS's VOM architecture, showing the relationship between VOM and other key components, including application execution, data management, system-level VO support, and security. *XOS-Cred* — a set of short-lived VO credentials that the VOM grants to a user application — connects the components. It has two parts: one, *XOS-Cert*, is public (including public key, VO identity, and VO attributes), whereas the other is a private part (such as a private key). Through XOS-Cred, a user application can establish trust relationships with grid entities, such as resource nodes or data management services. The system-level VO support mechanisms help enforce system-level resource usage control, accounting, and isolation.

Application execution management (AEM) services manage application execution,[2] transmitting XOS-Cert to resource nodes for authentication and authorization. The actual resource consumption feeds back to VOM through AEM so that VO-wide accounting is realized in real time. AEM also uses the VO policy service to enforce VO-wide resource usage control and filter out any resources that don't comply with VO policies.

XtreemFS is XtreemOS's distributed object-based file system.[3] It takes VO attributes (such as groups) from VOM to set up its access-control lists (ACLs). Together with XOS-Cred, ACLs enforce access control to data files.

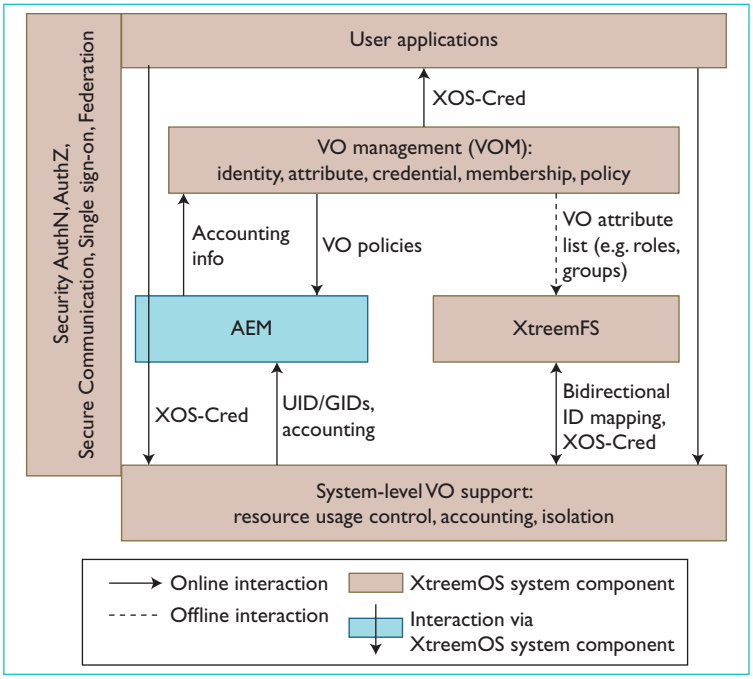


Figure 1. The XtreemOS VO management architecture. We show the interplay among user applications, XtreemOS services (the Application Execution Management and XtreemFS) and the node-local VO system support, as coordinated by the VOM services.

Security Services for VO Management

Each VO is associated with an actor, the VO manager, who runs all VOM services and has a public key certificate from a recognized certificate authority (CA). The certificate lets the VO manager issue VO credentials to users to access grid resources. The VOM architecture's security services include the following:

- The XtreemOS VO membership service (X-VOMS) validates the memberships of users who initiate a grid request. X-VOMS looks up user information, such as identity and attributes, in a VO.
- The identity service (IDS) generates and manages globally unique VO IDs and user IDs. The system's architecture assumes that resource nodes trust the VO manager. We achieve this by requiring all nodes to pre-install the manager's root CA certificate. Nodes can verify users' authenticity based on their XOS-Cert.
- The attribute service (AttrS) provides users with VO attributes, which let AEM services check against VO policies during resource selection, perform access control to resources and XtreemFS files, enforce system-level resource usage control, and let nodes map global IDs to system UIDs/GIDs.
- The credential distribution authority (CDA) issues users VO credentials for accessing grid-wide services and resources. XOS-Certs are represented using the X.509 certificate format. CDA holds the VO manager's public–private key pair so that it can issue signed XOS-Creds.
- The VO policy service (VOPS) provides policy-related services, such as policy information and decision points, to VOM so that resource access control is enforced not only at nodes, but also by VOM. Integrating policy decisions in VOM means that VO policies can be incorporated in job scheduling, resource negotiation, and VO-wide coordinated resource usage control.

CDA and VOPS offer interfaces to exter-

nal components, such as AEM and XtreemFS, whereas IDS, AttrS, and X-VOMS are the backend services for CDA. All XtreemOS VO-related services run in user space, and running as root isn't mandatory. Like all grid efforts, some XtreemOS services (such as the AEM service) need to run with enough capabilities to create user sessions on remote machines. We nevertheless minimize the presence of privileged processes and restrict their capabilities in order to reduce system vulnerability.

System-Level VO Support

The policies that a VO specifies, such as security, resource limitations, scheduling priorities, and rules on sharing resources by VO members, will be checked and ensured at resource nodes by the OS's local instance.

For the Linux OS kernel to enforce VO policies, we must let it deal with VO and VO users' identities, exploit this identity information in standard access-control mechanisms, and supply it to grid-aware system modules (such as XtreemFS).

Linux is unaware of VOs, and no high-level policy languages exist that encompass several systems. Currently, isolation and access control mainly rely on user accounts, process identities, and file permission bits, although the latest kernels support mandatory access-control extensions, including POSIX capabilities, file ACLs, and SELinux extensions.

To exploit these mechanisms, node-level VO support must provide mapping from VO-level identities and policies to local ones that Linux can fully recognize.

VO-customizable, dynamic mapping of grid users onto local accounts. We have integrated XtreemOS grid-user management into the Linux OS using Pluggable Authentication Modules (PAM),[4] Name Service Switch (NSSwitch),[5] and Kernel Key Retention Service (KKRS),[6] all of which are already present in standard Linux distributions. System administrators can develop and install custom PAM and NSSwitch modules without changing the kernel, while still remaining compatible with legacy applications.

Grid-user requests can then be mapped in different ways, depending on the VO, by choosing the appropriate PAM/NSSwitch modules. For compatibility, grid users can be mapped to anonymous local accounts, as often happens in Globus, or to specific, predefined accounts as in Grid User Management System (GUMS)[7] and Local Centre Authorization Service/Local Credential MAPping Service (LCAS/LCMAPS).[8]

In XtreemOS, we achieve scalable management of large VOs and grids through dynamic generation of local UID/GID on computing resources, according to the credentials stored in the XOS-Cred. The number of local accounts needed for executing processes in a resource node is bounded by the number of users simultaneously using the resource, regardless of the overall number of VOs and users per VO. Dynamic local identities also provide a degree of isolation among users, given that actual grid identities are concealed at the system level.

Interfacing to the grid authentication services. PAMs integrate multiple low-level authentication technologies (such as Kerberos and SQL-based authentication) into a common high-level API. Applications requiring authentication can be developed independently of the underlying authentication mechanism, which is embedded within specific PAMs. The PAM approach can be used to apply customizable policies in three phases of service execution: authentication, authorization, and session management.

XtreemOS exploits PAM technology to interface with Linux and grid authentication services. With the XtreemOS PAM module, grid users authenticate using their XOS-Cred certificate, and VO policies are enforced during authorization. Session management in XtreemOS-PAM implements dynamic management of local accounts, with automatic housekeeping of local files and processes, credential management, and name service updates.

User-space credential translation. NSSwitch, included in the GNU C library (libc), is a mechanism for intercepting queries to traditional Unix file-based information databases (such as password and group files), replacing them with other databases.

In XtreemOS, the NSSwitch module translates account-related information according to the PAM-established mapping. The information is maintained in databases accessible through a local account mapping service (AMS). By embedding account resolution into an NSSwitch module, we remain compatible with legacy ap-

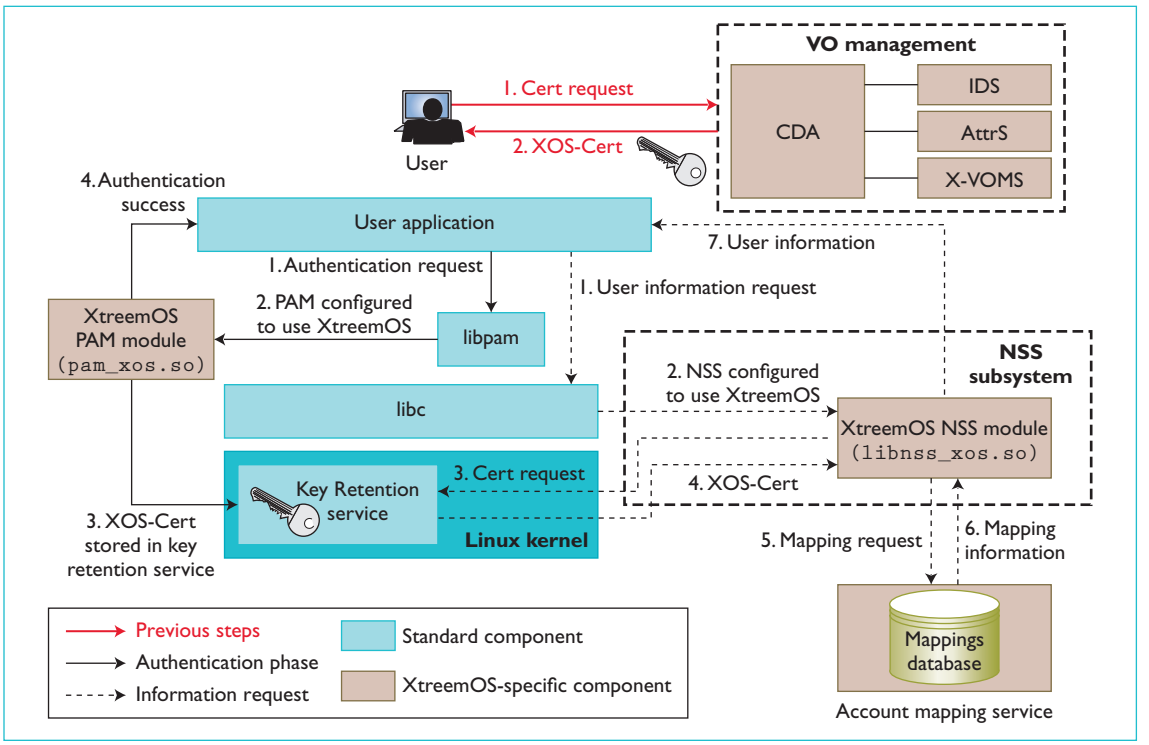


Figure 2. Name Service Switch (NSSwitch) and Pluggable Authentication Modules (PAM) modules used to authenticate users and map global and local accounts. Continuous arrows represent the successive phases of user authentication and resource authorization for a locally mapped user. Dashed arrows represent the interactions needed to retrieve from the VO services information about a user, and possibly to generate a new local mapping for a grid user.

plications, and we restrict access to global users' account information to authorized VO users.

Access control and logging. Mapping grid users to local accounts lets the XtreemOS system exploit all access-control mechanisms Linux provides. In particular, the KKRS is essential to XtreemOS; it caches authentication data related to a process within the kernel. Other kernel services, including file systems, can access this information and delegate operations to authenticated user-space applications.

During session initialization, XtreemOS-PAM stores the user's XOS-Cred in the kernel session keyring to be retrieved each time the global user credentials need to be used. PAM-aware services can then check user authorization and use the credentials, and local service auditing and resource usage accounting are possible.

Current Development

The first XtreemOS release is undergoing an integration phase and will be tested against a range of use cases selected from 14 scientific and commercial applications. Planned releases will be deployed on the Grid'5000 (https://www.grid5000.fr/) large-scale Grid testbed, which INRIA, a leading partner of XtreemOS, manages.

System-Level Prototype

We have implemented a local-level prototype of account mapping and user authentication using X.509 proxy certificates in a PAM and an NSSwitch module. The mappings are stored in a separate (local) database for grid users, independent from user databases.

Figure 2 shows the structure of the node-level prototype. Typically, a grid user, who is a VO member, would obtain an XOS-Cert from a VO manager and present it to the user application on a PAM-aware resource node in the VO. PAM would check the XOS-Cert for validity and store it in the KKRS associated with the user process. Thus, this process and its children can show that certificate to local and remote services.

Grid-Level Implementation

We have focused our grid-level implementation on CDA and VOPS because they provide external interfaces to other XtreemOS services.

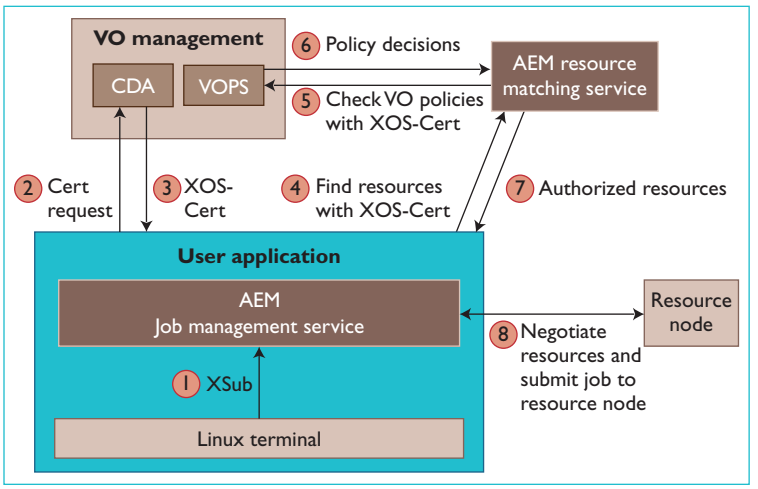


Figure 3. Grid-level implementation: the Securing Application Execution in XtreemOS. We show the main XtreemOS subsystems that are required to start a new application, their explicit interactions with VO remote management services, and all the interactions implicitly mediated by the local, system-level VO support.

Figure 3 illustrates the current grid-level implementation, in particular using CDA and VOPS to support AEM resource negotiation and job submission. Users can start using grid resources by typing the XSub command to what appears as an ordinary Linux terminal window. CDA provides XOS-Cert for a user to access grid resources. Note that the private key is "private" to the user, and the CDA doesn't know it. A user application can submit resource selection requests to the AEM Resource Matching Service, which authenticates the user by checking the XOS-Cert and fetches appropriate resources based on the request. VOPS ensures these resources' validity, including the usage, and a list of authorized resources is returned to the user application. Via the XOS-Cred and VOPS authorized decisions, the user application can negotiate resource allocation with nodes and subsequently execute jobs, assuming the negotiation is successful. The handling of XOS-Cert is then handed over to the node-level VO support mechanisms.

The CDA design references that of MyProxy (http://myproxy.ncsa.uiuc.edu) because of its online credential distribution capability and good provision of authentication choices (such as Kerberos and Grid Security Infrastructure). This strategy allows us the flexibility of interoperating with existing grid deployments.

We chose the Extensible Access Control Markup Language (XACML)[9] to implement VOPS because it's an XML standard for access-control policy language, and we can extend it with additional attributes and policy combining algorithms, an enabling feature for managing dynamic VOs.

Discussion

The XtreemOS approach of managing a whole grid providing the abstraction of a single computing platform draws ideas and shares some implementation choices with state of-the-art results in the VO field. The Virtual Organization Membership Service (VOMS)[10] is an important reference implementation for XtreemOS because it's a popular approach for utilizing users' VO attributes (such as group and role) in node-level account mappings and dynamically allocating local accounts to grid users. However, the number of dynamic accounts is bounded by the size of a predefined pool of accounts configured on each node. When the number of concurrent users having the same VO attributes exceeds that limit, VOMS "recycles" pool accounts, which means that users have to share the same account.

Our approach improves on VOMS by removing the need to have predefined accounts and to limit their number. Necessary UIDs and GIDs are created on-the-fly, deleted after use, and, thanks to NSSwitch, don't appear in traditional Linux account files. Grid user and group accounts can be "invisible" to local users. This approach also provides a maximum degree of compatibility to grid-unaware applications.

The Community Authorization Service (CAS)[11] pushes VO policy decisions to nodes to support VO-wide authorization. Nodes delegate the right of access to their resources to the CAS server; thus they only act as policy enforcement points. This is fundamentally different from the approach taken by VOMS and XtreemOS, in which the ultimate right of access to resources remains with resource owners.

Legion[12] (http://legion.virginia.edu) uses an object-based programming model to realize its vision of a single virtual machine to mask the underlying complexity of a wide-scale distributed environment. XtreemOS doesn't try to hide the fact that users and resources reside in a heterogeneous and distributed environment. As opposed to the Legion programming model, XtreemOS aims to equip existing applications with grid capabilities with the minimum amount of refactoring.

Grid virtualization[13] is another strand of work related to XtreemOS that addresses some of

our concerns (such as strong isolation). However, it doesn't provide all the answers, given that not all applications require strong isolation, and user account-level isolation is often sufficient because it provides the capability for isolation and sharing. Although the current state of virtualization technologies makes it difficult to share data between users within a VO and across VOs, the XtreemOS VO management architecture, on the contrary, can easily be extended to use virtual machines or process containers to provide stronger isolation properties.

XtreemOS aims to provide an infrastructure combining the scalability and resource management VOs offer with the efficiency of an OS. The approach of providing core grid services with minimal extension to Linux ensures compatibility with current applications and adaptability to different collaboration models.

We've started developing a systematic threat-analysis framework to ensure the quality of XtreemOS design and software and provide testable assurance to enhance the likelihood that XtreemOS gains more widespread acceptance. XtreemOS is engaging the Linux and grid communities in an open source development in several ways, such as by having a Sourceforge presence and by packaging and disseminating XtreemOS releases through major Linux vendors, including Mandriva (www.mandriva.com) and Redflag (www.redflag-linux.com). Involving the wider community ensures response to its needs and provides a sustainable path to take this XtreemOS into the future.

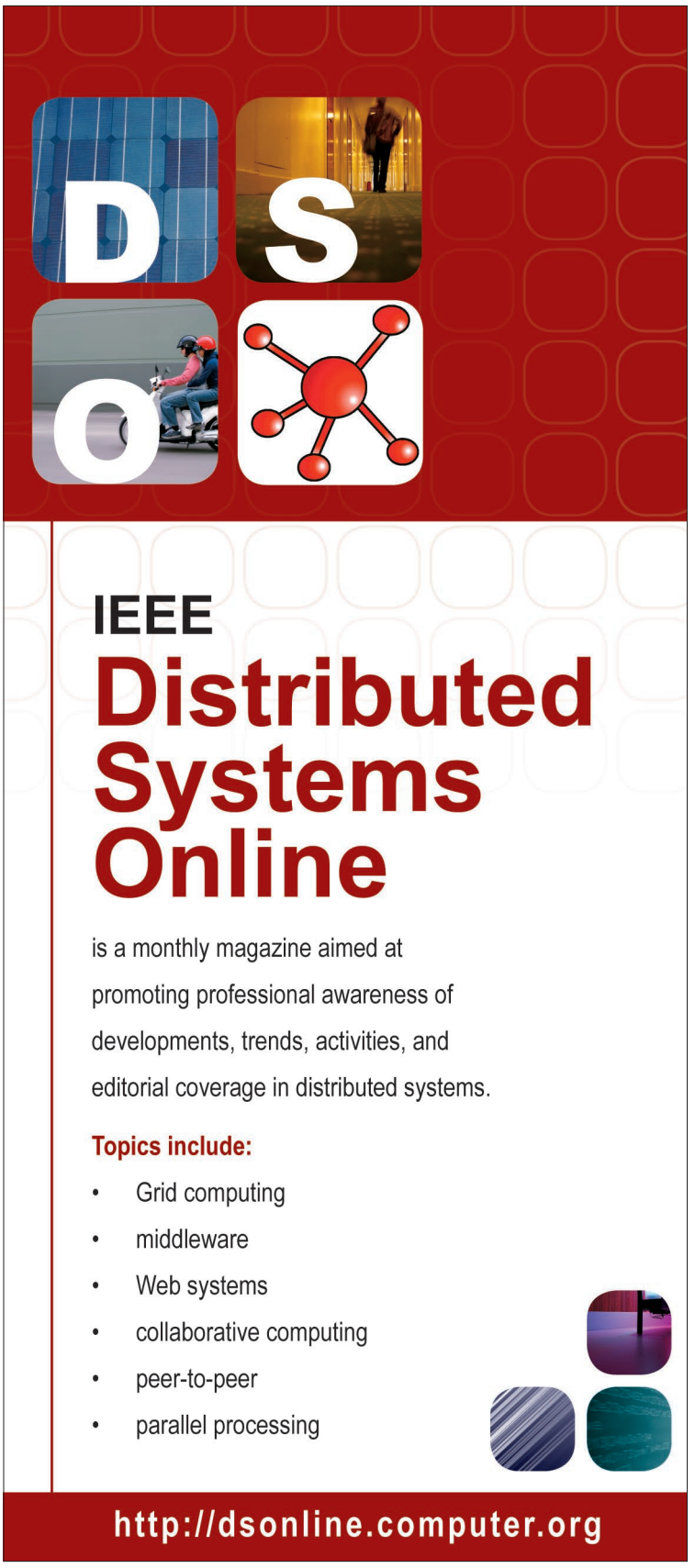
References

1. XtreemOS WP 4.2, "Requirements Capture and Use Case Scenarios," Jan. 2007; www.xtreemos.org/publications/public-deliverables.
2. XtreemOS WP 3.3, "Requirements and Specification of XtreemOS Services for Application Execution Management," Nov. 2006; www.xtreemos.org/publications/public-deliverables.
3. XtreemOS WP 3.4, "The XtreemOS File System – Requirements and Reference Architecture," Dec. 2006; www.xtreemos.org/publications/public-deliverables.
4. V. Samar and R.J. Schemers III, "Unified Login with Pluggable Authentication Modules (PAM)," *Open Software Foundation RFC 86.0*, Open Software Foundation, Oct. 1995.
5. Free Software Foundation,, "GNU C Library Reference Manual," www.gnu.org/software/libc/manual/.
6. A. Kumar and S. Chopdekar, "Get Started with the Linux Key Retention Service," Apr. 2007; www.ibm.com/developerworks/linux/library/l-key-retention.html.
7. R. Baker, D. Yu, and T. Wlodek, "A Model for Grid User Management," 2003; www.citebase.org/abstract?id=oai:arXiv.org:cs/0306063.
8. R. Alfieri et al., "Managing Dynamic User Communities in a Grid of Autonomous Resources," *Proc. Computing in High Energy and Nuclear Physics*, Stanford Linear Accelerator Center, Mar. 2003; www.slac.stanford.edu/econf/C0303241/proceedings.html.
9. The OASIS Consortium, "eXtensible Access Control Markup Language (XACML) 2.0: Specification"; http://docs.oasisopen.org/xacml/2.0/accesscontrol-xacml-2.0-core-spec-os.pdf.
10. The Enabling Grids for E-SciencE (EGEE) Team, "EGEE User' Guide – VOMS Core Services," 2005; http://egee.cesnet.cz/en/voce/voms-guide.pdf.
11. The GLOBUS Team, "CAS in Globus 4.2"; www.globus.org/toolkit/docs/development/4.2-drafts/security/cas/.
12. A. Grimshaw et al., "Architectural Support for Extensibility and Autonomy in Wide-Area Distributed Object Systems," http://legion.virginia.edu/papers/CS-98-12.pdf.
13. K. Keahey et al., "Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid," *Scientific Programming J.*, special issue on dynamic grids and worldwide computing, vol. 13, no. 4, 2005, pp. 265–275.

Massimo Coppola is a researcher with the High Performance Computing Lab at the Institute of Information Science and Technologies (ISTI/CNR), Italy. His research interests include parallel computing architectures, parallel programming models, and high-performance and data-intensive applications, including data mining. Coppola has a master's and a PhD in computer science from the University of Pisa. Contact him at massimo.coppola@isti.cnr.it.

Yvon Jégou is an INRIA researcher and is working in the PARIS research project of INRIA-Rennes Bretagne Atlantique, France. His research interests include architecture, operating systems, and compilation techniques for parallel and distributed computing. Contact him at yvon.jegou@irisa.fr.

Brian Matthews is the leader of the Information Management Group in the E-Science Centre of the Science and Technology Facilities Council, UK. His research interests include formal software engineering methods, metadata, Web and grid systems, trust management,



and virtual organizations. Matthews has a PhD in computing science from the University of Glasgow. He is a member of the British Computer Society. Contact him at b.m.matthews@rl.ac.uk.

Christine Morin is a senior researcher at INRIA Rennes Bretagne Atlantique, working in the PARIS project team, and the scientific coordinator of the XtreemOS project. Her research interests include operating systems, distributed systems, fault tolerance, and cluster and grid computing. Morin has an engineering degree from the Institut National des Sciences Appliquées (INSA), of Rennes, France, and her master's and PhD in computer science from the University of Rennes 1. She is a member of the ACM and the IEEE. Contact her at christine.morin@inria.fr.

Luis Pablo Prieto is a research engineer at Telefónica I+D (Telefónica Investigación y Desarrollo) in Spain. His research interests include systems for personal communications, mobile ad hoc networks, and mobile grids. Prieto has a telecommunications engineer degree from the University of Valladolid. Contact him at lpps@tid.es.

Óscar David Sánchez is an R&D project manager at INRIA-Rennes Bretagne Atlantique, where he coordinates the XtreemOS project's technical activities. His research interests include virtual organizations, security, and grid architectures. Sánchez has an MSc in telecommunications engineering and an MPhil in computer science from Universidad Politécnica de Valencia, Spain. Contact him at oscar.sanchez@inria.fr.

Erica Y. Yang is a senior research associate in the E-Science Centre of Science and Technology Facilities Council, UK. She leads the security and VO management research strands in XtreemOS. Her research interests include security and fault tolerance in distributed systems, grids, and P2P networks. Yang has a PhD in computer science from the University of Durham, an MSc in high-speed networks and distributed systems from Oxford Brookes University, and a BSc. in applied mathematics from South China University of Technology, China. Contact her at y.yang@rl.ac.uk.

Haiyan Yu is an associate professor of the Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS). His research interests include distributed systems and high-performance and grid computing techniques. Yu has a PhD in systems engineering from the Beijing JiaoTong University of China. He is a member of the IEEE and the ACM. Contact him at yuhaiyan@ict.ac.cn.