More detailed description of Duke's AD use case (Delegated Directory Administration)

Use case detail:
For a number of years, our central IT organization has operated a disconnected AD forest for the campus, but since the creation of user objects in the AD has always been a manual process on the part of the central IT organization, adoption among departments (both academic and administrative) on campus has been very low. The enterprise AD has offered few advantages for departments over departmentally-run forests, but as a shared resource, has brought with it limitations not present with departmental forests.
Now that the central AD forest has been integrated into the enterprise identity management "cloud", departmental interest in the resource is rapidly exploding. Departments want to transfer their distributed identity tracking responsibilities for their traditional University affiliates (staff, students, faculty, emeriti, alumni, enterprise affiliates) to central IT in order to take advantage of centralized IDM services. They wish to retain the ability, however, to manage their departmental desktops and servers themselves, and the ability to manage the profiles, policies, and in some cases, attributes on their departmental users which pertain to departmentally-provided resources (such as faculty and graduate student Windows server home directories). They also need to retain the ability to create and modify user objects for users not recognized by the institution as affiliates, but recognized by their departments as "persons of interest" -- collaborators, family members of faculty, departmental administrative and test user accounts, etc.
The primary needs we're trying to address are:

(1) Our Oracle IDM is managing basic attributes (givenName, sn, cn, sAMAccountName, ou, eduPerson*, etc.) in the AD based on information collected from our ERPs (SAP for employees and PeopleSoft for students). Where an attribute on a managed user is "owned" by the IDM, we want to avoid any delegated administrative access to the attribute. There are attributes in the AD schema (some Microsoft native, some extensions we/ve added to the schema) which primarily connect users to resources provided at the department level (eg, homedirectory, which for staff, faculty, and graduate students in most departments, will refer to a departmental fileserver of some sort). In those cases, we need to provide a level of distributed delegation of management to allow departmental staff to manage those attributes while reserving control over other user attributes for the IDM.

(2) Our departmental admins also need the ability to freely create, modify, and delete other objects which are not to be managed by the central IDM facility. Most of these will be resource objects (computers, etc.) but some may be user objects representing either special administrative user instances (eg., a "superuser" account or some AD-dependent software product installed within a single department) or people whose affiliation with the University is not strong enough to warrant their vetting as University affiliates, but to whom the department wishes to extend some level of authenticated access to departmental resources in the Active Directory. To facilitate this, we plan to "split" our AD tree into two main OU branches -- one containing only user objects that are primarily managed by the enterprise IDM, where departmental admins may have limited rights delegated to them, but where significant restrictions will always apply, and one containing only departmentally-managed objects (users and resources), where individual administrators will have full object-level access within the scope of specific structural OUs. We anticipate the OU hierarchy in both branches being the same, so that where there exists an OU, for example, representing a department in the "managed" branch of the AD tree, there will also be a matching OU in the "departmental" branch of the AD tree.

(3) There are cases in which a one-to-many relationship exists between a person and multiple departments or orgunits -- faculty may have appointments in mulitple departments, and staff may work across orgunit boundaries. In these cases, we would like to provide the IT admins in all the relevant departments limited rights to manage attributes on those user's objects in the AD without granting them rights to user objects they have no business need to access. The AD OU structure is of little help, since it has no means to represent one-to-many relationships of this sort, so we anticipate needing to represent multiple orgunit associations through groups in the AD. The ACLing mechanism in AD is not well-suited to the kind of permission management we need, however -- AD doesn't seem to implement ACL application based on attributes of target objects, so while members of a group can hold a privilege as a result of their membership, the privilege cannot be scoped to only apply to objects that are members of an object group. One can grant members of the Frisbee team group access to some resource, but one can't grant an individual access to all resources with the "isFrisbee" attribute set, nor all resources that are members of the "flying disk" group. We'll need some means for tracking one-to-many and many-to-many relationships outside the AD and for projecting them as many-to-one or one-to-one privileges into the AD.

(4) The rights any individual has need to be driven both by the individual's identity (in the case of explicit grants to specific users) and by the individual's business roles and position within the orgunit hierarchy. If an individual's roles change or an individual leaves or moves between orgunits, we need the rights the individual has to change accordingly and automatically. Becoming an IT admin in a department should automatically cause one to accrue certain rights, while leaving the position should automatically cause some rights to be removed.

(5) Rights can and should inherit down the orgunit hierarchy. An IT admin in a position scoped to an entire school should receive rights that apply not only to leaf objects rooted at the school level, but also to leaf objects rooted at the department and sub-department levels below the school, but only to objects contained within the directory subtree rooted at the school.

(6) There are likely to be occasional exception cases of at least two forms:

(a) An IT admin whose user object resides in a given OU may need to have administrative privileges in an OU that's neither his own nor a sub-OU of his own. We occasionally have part-time IT admins working in multiple orgunits, and although they'll themselves be positioned in only one OU, they'll need to have administrative rights associated with their connection to other orgunits.

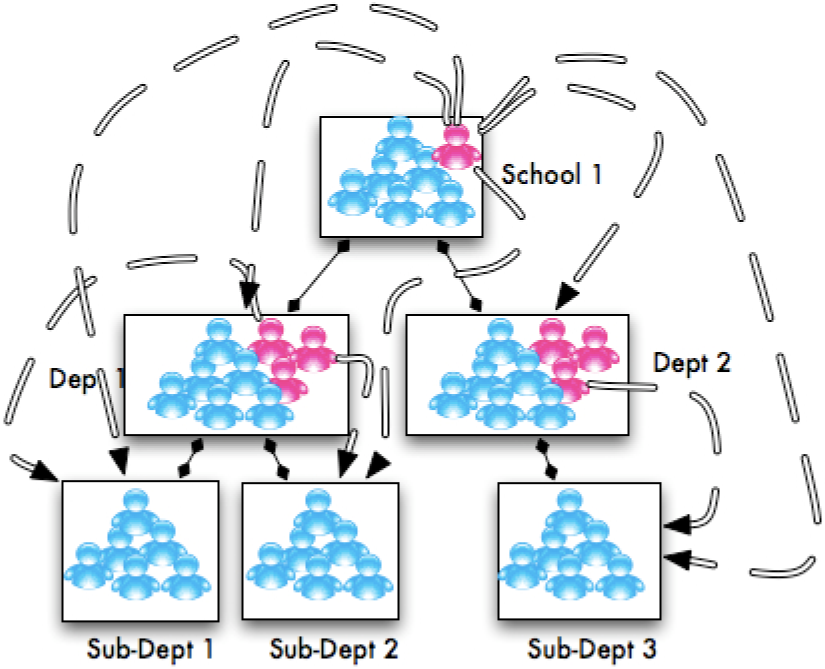
(b) An IT admin whose user object resides in a high-level OU (say, at the level of a school or division) may need to be explicitly denied rights in a sub-OU further down the org unit hierarchy. This case arises in situations where a department or unit may "break away" from the IT organization within its division or school and take over its own administrative responsibilities internally. It's more common in the academic sector than the business sector of the institution, and has happened only occasionally over the years, but it does happen. Currently, I know of only one case on campus like this -- as you might guess, the Computer Science department wants to avoid using the College's IT resources and prefers to strike out on its own, but they're still interested in leveraging the central IT shop's IDMS services, including AD.

Our departmental admins would prefer, by their own admission, to continue doing business as they have in the past, using Microsoft tools to manage data directly in the AD. We in central IT would like to ensure that AD ACLs are in place to limit IT admin's privileges to those policy dictates, but we would be equally happy with a solution that requries departmental admins to use another set of tools to manage their user's attributes. Managing privileges through an external system, so long as the external system can project changes in near-realtime into the AD or can interact with the campus IDM in a fashion that allows it to project changes in near-realtime into the AD, would be perfectly acceptable.

Initial Use Case Decomposition:
Multiple subjects, identified based on authoritative source data about their business roles, need to be granted attribute-level access privileges (for both reading and writing) within the AD. Access privileges need to be scoped based on authoritative source data about the subjects' positions within the orgunit hierarchy, and based on authoritative source data about the targets' positions within the orgunit hierarchy. Subjects with a particular business role (IT admin), need to be granted specific rights to manipulate AD objects within structural OUs in the directory that match their own OUs (eg., an admin in OU1 should receive rights scoped to objects within OU1).

Additionally, those same subjects need to be granted more extensive rights, scoped in the same fashion, within a different portion of the directory hierarchy that maps in similar fashion the orgunit hierarchy of the institution. The same rules apply regarding how subjects acquire and lose rights and how subjects' rights are scoped within the directory tree, but the specific rights in this second case are more extensive (and include full read/write/create/delete rights on objects within this alternate hierarchy).

White and black list capabilities are needed as an overlay to inferred rights based on orgunit and business role, since some relationships exist that contravene the rule of simple, hierarchical inheritance of rights. In this case, the number of white or black-listed cases is expected to be extremely small, but not vanishingly so.



In the depiction above, there is a single IT admin in School 1 who has rights over user objects in the school and its five subsidiary deaprtments and subdepartments. There are three admins in each of the two departments (Dept 1 and Dept 2) who hae rights both over their own departmental users and over the users in their respective sub-departments. No admins are positioned in the subdepartments, so the only individuals with rights over users in those subdepartments are the admins in the parent departments.
The depiction above does not include examples of any assignment of rights outside the normal orgunit hierarchy, nor examples of any denial of inherited rights , but both are requirements in the overall case we're hoping to address.