PROGRESS® DataDirect®

ADDRESSING SARBANES-OXLEY COMPLIANCE

INTRODUCTION

For IT departments in public companies the biggest challenges today are implementing the required IT policies and infrastructure to comply with the Sarbanes-Oxley Act of 2002 (SOX).

The Sarbanes-Oxley Act of 2002 (SOX) was passed by Congress to reform the accounting practices, financial disclosures and corporate governance of public companies. Among other things, SOX requires the CEO and CFO of public companies to personally certify that financial statements are accurate.

Section 404 of SOX requires that management perform an assessment of internal controls over financial reporting and obtain attestation from external auditors, on an annual basis. For most companies, the deadline for compliance with section 404 was November 15, 2004.

For section 404, organizations are expected to use an accepted framework to establish appropriate internal controls. The SEC specifically cites the framework developed by the Committee of Sponsoring Organizations of the Treadway Commission (COSO). The COSO framework makes general references to IT controls but is not a specific IT framework. The IT framework that is considered most closely aligned with COSO was developed by the IT Governance Institute and is known as COBIT (ControlObjectives for Information and Related Technology). COBIT sets forth specific IT control objectives, several of which relate directly to identity and access management.

This document deals with the identity and access management-related issues in *IT Control Objectives for Sarbanes-Oxley*, as published by the IT Governance Institute. It is not intended to be a complete discussion of the COBIT framework.

THE CHALLENGE OF MAINTAINING A CONSISTENT LEVEL OF CONTROL

According to the IT Governance Institute's document, IT Control Objectives for Sarbanes-Oxley, controls over information technology systems should include access controls over programs and data:

> "Access controls over programs and data assume greater importance as internal and external connectivity to entity networks grows. Internal users may be halfway around the world or down the hall, and there may be thousands of external users accessing, or trying to access, entity systems. Effective access security controls can provide a reasonable level of assurance against inappropriate access and

HIGHLIGHTS:

- Discover the scope of the data integration challenge
- Enforce consistent access control across multiple touchpoints
- Implement Fine Grained Secure Access Server

PROGRESS

unauthorized use of systems. If well designed, they can intercept unethical hackers, malicious software and other intrusion attempts. Adequate access control activities, such as secure passwords, Internet firewalls, data encryption and cryptographic keys, can be effective methods of preventing unauthorized access. User accounts and related access privilege controls restrict the applications or application functions only to authorized users that need them to do their jobs, supporting an appropriate division of duties. There should be frequent and timely review of the user profiles that permit or restrict access. Former or disgruntled employees can be a threat to a system; therefore, terminated employee passwords and user IDs should be revoked immediately. By preventing unauthorized use of, and changes to, the system, an entity protects its data and program integrity."

A typical enterprise has a number of applications it uses for running the business. These include ERP, HR, CRM, and others. Because of acquisitions and mergers, a single company may be running different applications for the same purpose – for example an ERP system from Oracle and from SAP—each from a different vendor and potentially storing data in a different database.

Each of these applications manages security as to who can access what. As long as the users are accessing the application from the vendor's front-end tools, all the access rights that have been configured within the application are enforced. But in many cases third-party applications that augment or replace the application's functionality are used to access the data.

For example, you decide to build custom analytics and reporting solutions using Brio. And you also deploy web applications. These applications typically go directly to the underlying rows of data in the database. How do you maintain the ability to use these third-party or in-house applications and at the same time enforce a consistent level of access control no matter how the data is accessed?

ENFORCING ACCESS CONTROL ABOVE THE DATABASE LAYER

One alternative is to make the lowest layer in the system secure to the required level. For many applications this means securing the database such as Oracle or SQL Server. The granularity required to comply with SOX and other regulations is not easily implemented and managed using the database features. And since an enterprise will have many different kinds of databases (Oracle, SQL Server, DB2, home-grown, etc), the same level of access control within the database cannot be achieved. Also, the management of taking a set of policies for different classes of users and mapping it to database level settings becomes a daunting task.

A better alternative is to force all third-party and in-house applications to access data sources through a middle layer that enforces access control above the database layer. In this write up we refer to this layer as Fine Grained Secure Access Server (FGSAS) (Figure 1). The applications themselves would continue using their built-in access control mechanism but all other access would be through the FGSAS that is developed in-house or purchased. Ideally the FGSAS can be configured

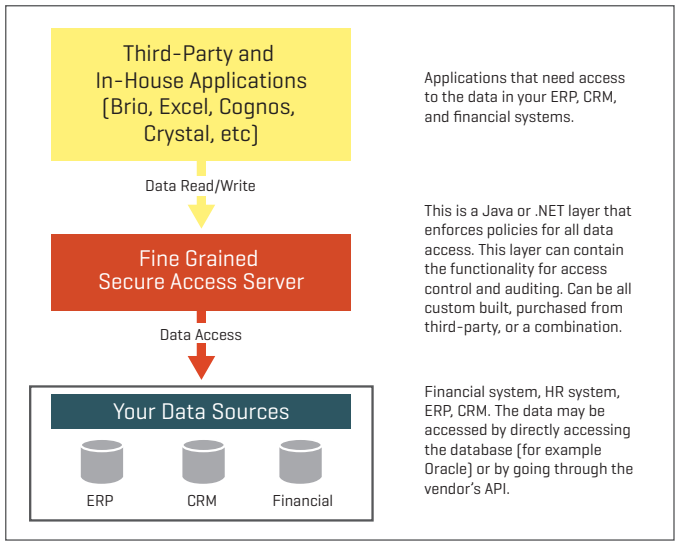

Figure 1: *Accessing Data through Fine Grained Secure Access Server*

with a set of policies based on a user's role without concern for the underlying data source.

What interface should this layer expose to the applications? It can expose business objects (such as Employees or Customers or Purchase Orders) that can be used from Java and/or .NET applications. Or it can directly expose tables that exist in the underlying databases. No matter what interface is exposed, unless it provides a SQL look and feel with an ODBC interface, the third-party applications used for reporting and analytics will not be able to access the data through your FGSAS. Typically an organization has many business intelligence, analytics, and reporting applications that access data from the various systems by directly accessing the underlying Relational Database Management System (Oracle, DB2, SQL Server, Sybase, and others). This connection is done using the Microsoft ODBC compliant drivers.

This is where DataDirect® OpenAccess™ comes in. OpenAccess allows the fine grained secure access server to be exposed as a virtual SQL database with an ODBC and JDBC compliant interface that looks and feels like the Oracle or SQL Server interface the client applications were developed to access. The client applications issue SQL queries to the OpenAccess-enabled server, which in turn uses the FGSAS to perform the data access.

HOW TO QUICKLY BUILD CENTRALIZED ACCESS CONTROL

OpenAccess provides the framework and pre-built components to quickly allow the implementation of a virtual SQL layer with ODBC and JDBC support over any data access layer using C, C++, Java, or .NET development environment (Figure 2). For example, to integrate a Java based FGSAS, all that is required is the implementation of glue code in Java (< 500 lines of code) to tie in the FGSAS to the OpenAccess SQL engine to support schema and data access requests. OpenAccess includes client/server support to allow the ODBC driver and the server to reside on separate systems. OpenAccess is supported on IBM AIX, Sun Solaris, Microsoft Windows, Linux, HP HP-UX, HP OpenVMS, HP Tru64, SCO Unix, IBM OS/390, and others.

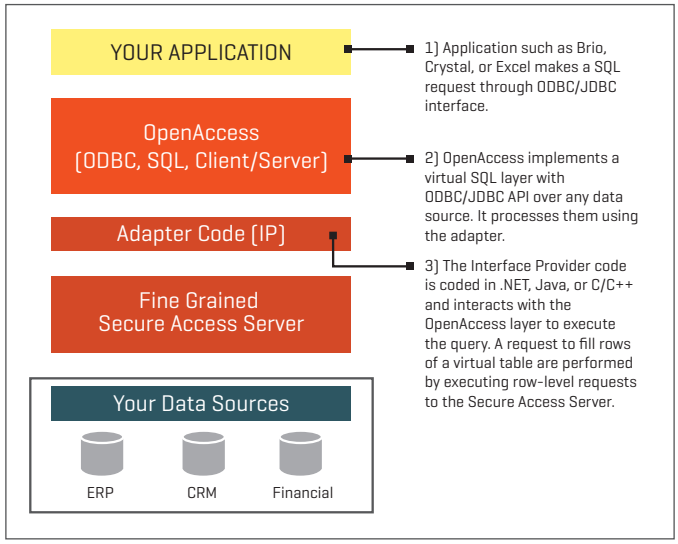

Figure 2: *OpenAccess Based Solution*

YOUR DEVELOPMENT EFFORT

1. Design and code the adapter code in either C, C++, Java, or .NET (14 days)
2. Do your QA (4 days)
3. Package up for distribution (2 days)

   Expected time of completion: **20 man days**

CONCLUSION

The ability to continue using ODBC/JDBC and SQL with a fine grained secure data access layer between the applications and the data sources gives enterprises the flexibility to build a centralized access control mechanism without breaking the existing applications. OpenAccess allows your custom code to appear as a virtual SQL database with an ODBC/JDBC API that is compatible with hundreds of applications in use today. Use of OpenAccess allows you to implement an enterprise quality custom ODBC driver by leveraging the OpenAccess platform, which includes 99% of what you need and allows you to code in the language of your choice to tie in the specific data source.

WORLDWIDE HEADQUARTERS

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA Tel: +1 781 280-4000 Fax: +1 781 280-4095 On the Web at: www.progress.com

Find us on facebook.com/progresssw twitter.com/progresssw youtube.com/progresssw

For regional international office locations and contact information, please go to www.progress.com/worldwide

Rev. 9/14