Visual Studioで構築するエンタープライズシステム

Application Architecture for .NETの利用例

第6回 セキュリティ

株式会社CSK
eソリューション技術部
中垣 健志
NAKAGAKI, Kenji

Level

1 2 3 **4** 5

Technology Tools

- ☑ Visual Basic
- ☑ Visual C#
- ☐ Visual C++
- ☑ SQL Server
- ☐ Oracle
- ☐ Access
- ☑ ASP.NET
- ☐ Other:

Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、http://www.shoeisha.com/mag/windev/からダウンロード可能です。

はじめに

都心に並ぶビルの最上階には、レストラン街があります。N君は、ときどき先輩のSさんと和風パスタを食べに出かけます。一回り以上も年は離れているのですが、気さくにいろいろな話をしてくれるのでSさんとの昼食はN君の楽しみのひとつ。今日はエレベーターの中で車の塗装の話になりました。Sさんいわく、見た目はほとんど変わらなくてもメーカーや車のクラスによって塗装の回数が違うらしいのです。

「へー。見た目は同じなのに、仕上げに差があるんですね」
「そう。車って意外と傷がつく機会が多いけれど、塗りが多ければ小さな傷なんかはほとんど目立たなくできるんだ」

カタログではきれいな車体も、実際に乗り始めると手を抜いたところがあらわになってしまうのは、アプリケーションも同じです。開発時にはうまく動いているアプリケーションも、本番環境では思わぬ使い方をされたり、あるいは悪意を持って不正利用されることがあるのです。そのような場合でもアプリケーションを正しく動かすためには、顧客の求める業務上の機能以外にも考えなければならない機能があるのです。今回は、そういった「非機能要求」に目を向けてみましょう。

非機能要求を見落とさない

システムを開発するにあたって顧客のニーズを聞き出すために「ヒアリング」という手法をとることがあります。これは、システムを必要とするエンドユーザーや情報システム部のメンバーから、システムに求める要件を聞き出す手法です。ヒアリングはシステムに必要な

要件を洗い出すために有効ですが、ヒアリングのみに頼ってシステム要件を完成させることは危険です。なぜならば、システムが必要とする機能の中には、顧客が“暗黙のうちに”期待しているものがあるからです。

その中でも特に気をつけなければならないものとして、今回のテーマである“セキュリティ”が挙げられます。セキュリティに関する機能が十分に実装されずシステム内の情報が流出してしまった場合、多くの損失が発生するのに加え、システムを運営する顧客自身の信頼も失われてしまいます。このような決定的なリスクがあるにもかかわらず、顧客側からシステムに対してセキュリティの要求がしっかりと出されることはまれです。なぜかと言えば次のような理由が挙げられるでしょう。

・顧客が、セキュリティや運用管理に関する正確な知識を持たない
・顧客は、まさかセキュリティ対策の施されていないシステムを納品されるとは思っていない

Application Architecture for .NET（以下AAfN）では、これらの問題をまとめて扱うために「セキュリティ」「運用管理」「通信ポリシー」というレイヤーを設けています。

◆セキュリティ

一口にセキュリティを確保する、といっても何に気をつければよいのか対策をとるべき範囲はとても広いものです。AAfNではこのセキュリティというカテゴリを次の5つに分類しています。

①認証：安全な形で身元を確認すること
②承認：認証されたユーザーのアクセス許可を管理すること
③安全な通信：通信される情報を盗聴や改ざんから守ること
④監査：不正利用がないかどうか、システムの利用状況の記録をとること
⑤プロファイル管理：保存されているユーザー情報を安全に管理すること

それぞれのカテゴリについてどれだけのコストと手間をかけるのかというポリシーを、情報の価値によって決定します。セキュリティの実現方法は、たいていの場合.NETによって用意されているか、あるいは市販品やデファクトスタンダードとしてあらかじめ準備されているものがほとんどです。特にセキュリティに関する実装は自作のものを使うことは避け、広く性能や安全性の実証されているものを使うべきです。

ここでは5つのカテゴリの中から、よく使われる認証、承認について具体的な実装手段を見ていきます[注1]。

◆認証

認証とは、アプリケーションに対して身元を明らかにする作業を指します。パスワードやクライアント証明書などによって利用者の身元が明らかにされたときに、.NETアプリケーションの中では、ユーザーを一意に識別する「アイデンティティ」が用意されます。アイデンティティはアプリケーションの中で重複しない名前を持っています（例：'nakagaki'）

ASP.NETでは、3通りの認証が用意されています（表1）。

Soarアプリケーションでは実行環境になるべく制約を設けないようにしたいので、「フォーム認証」を採用することにします。

◆認証チケット

ASP.NETでは認証がされていない状態で承認が必要なページを開こうとすると、自動的にログインページへと画面が遷移します。ASP.NETでは認証がされているかどうかを、どのように判断しているのでしょうか？

その答えは「認証チケット」です。ASP.NETは認証チケットがクライアントのクッキーに保存されているか

注1）「認証」と「承認」は、英語ではそれぞれ「Authentication」と「Authorization」と言い、日本語／英語どちらの場合も言葉の雰囲気が似ているので、この2つを同一視してしまいがちです。明確に違いを説明できるようにしておきましょう。