EDGE
Worldwide Community for COOL Application Developers

Applying COOL:Security Services

Tony R. Pierce

ActiveDevelopment Group, Inc.

Session PU 104

Monday, April 30, 2001

Agenda

- What is Application Security?
- The Service-based Software Architecture
- Application Security Data Elements
- Security Administration
- Applicability for Multiple Client Types
- Demonstration and Questions

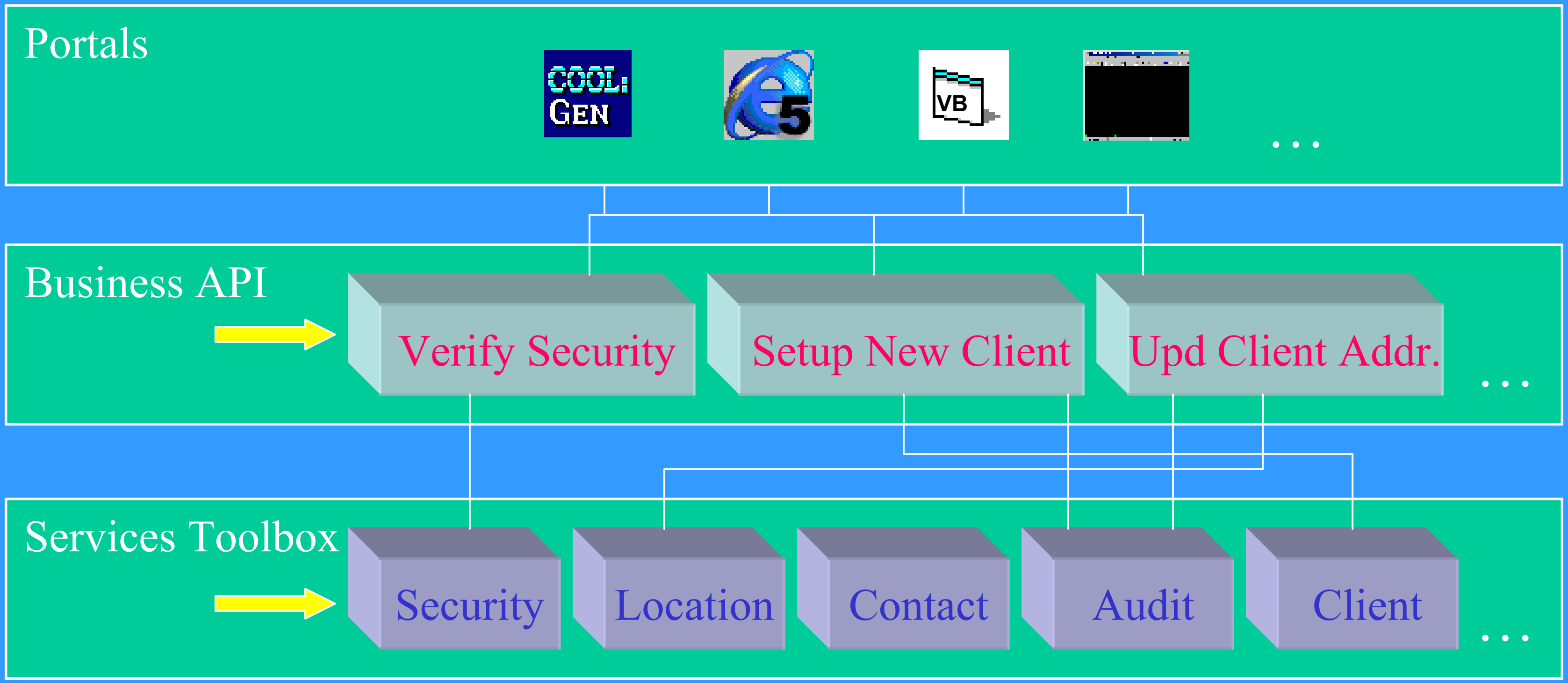
EDGE
Worldwide Community for COOL Application Developers

What is Application Security?

- A mechanism for:
  - Authenticating software users
  - Applying discrimination in terms of software functions through authorization
  - Providing individualized portals to software and information assets

EDGE
Worldwide Community for COOL Application Developers

Service-based Software Architecture



Portals

COOL:GEN

. . .

Business API

Verify Security

Setup New Client

Upd Client Addr.

. . .

Services Toolbox

Security

Location

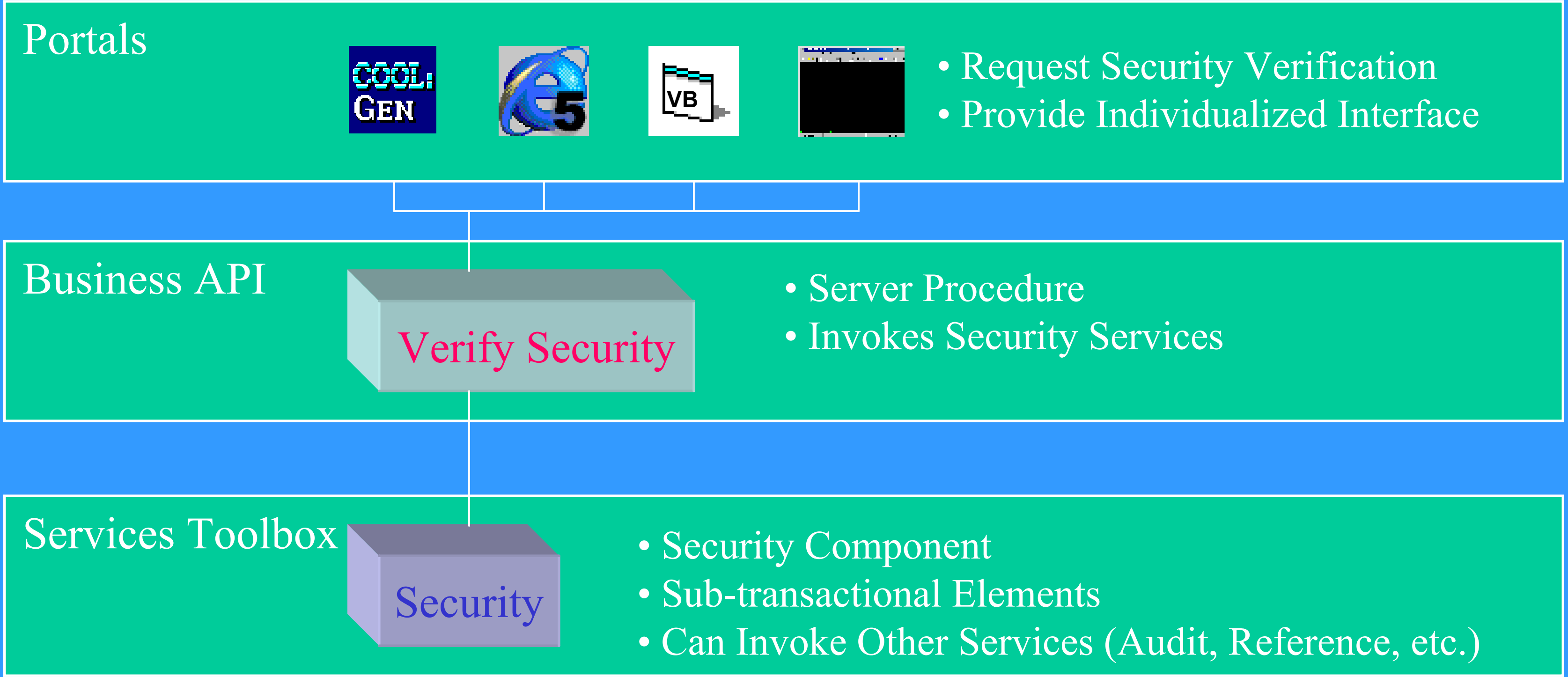
Contact

Audit

Client

. . .

Service-based Security Verification



Portals

- Request Security Verification
- Provide Individualized Interface

Business API

Verify Security

- Server Procedure
- Invokes Security Services

Services Toolbox

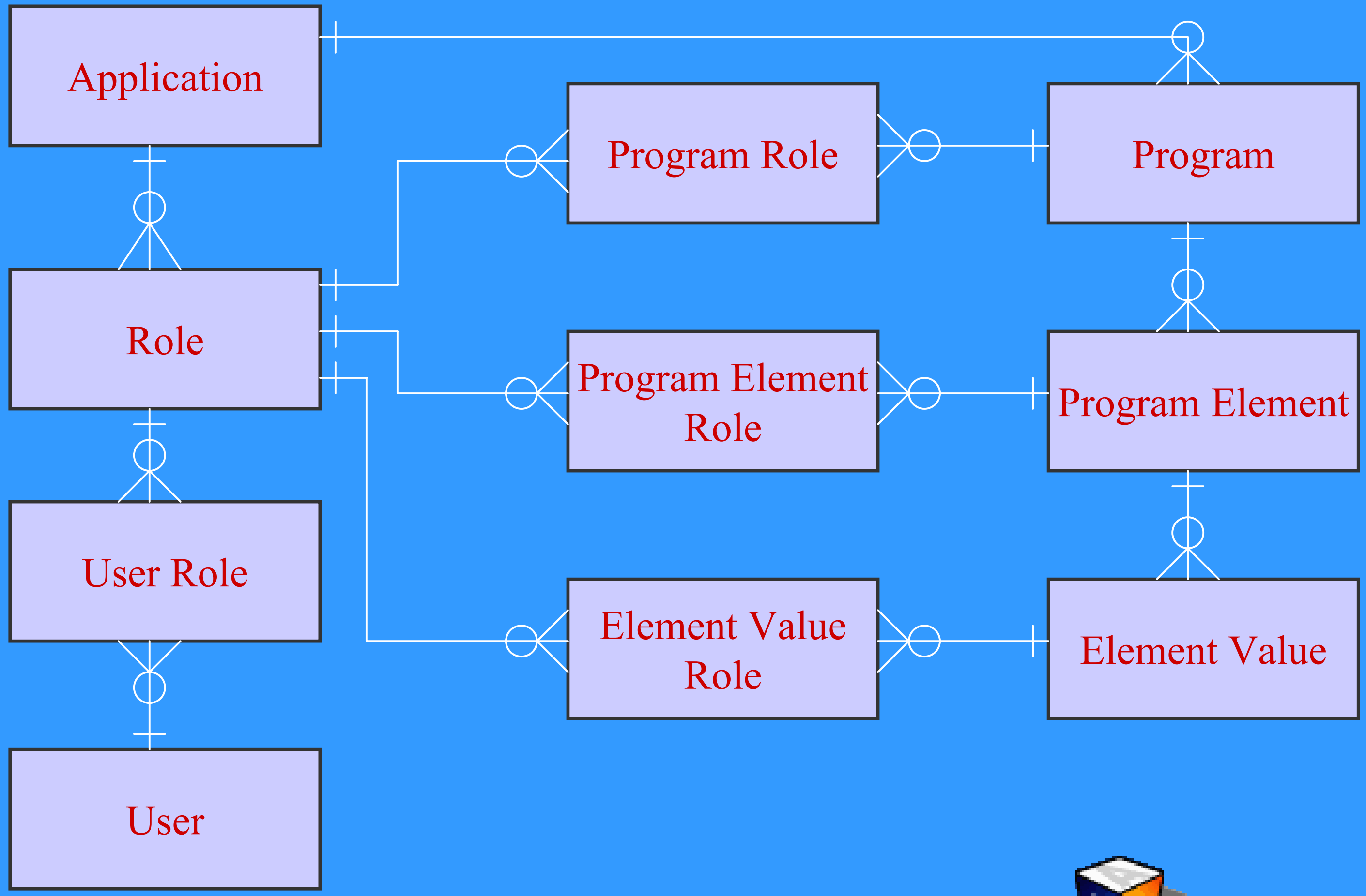
Security

- Security Component
- Sub-transactional Elements
- Can Invoke Other Services (Audit, Reference, etc.)

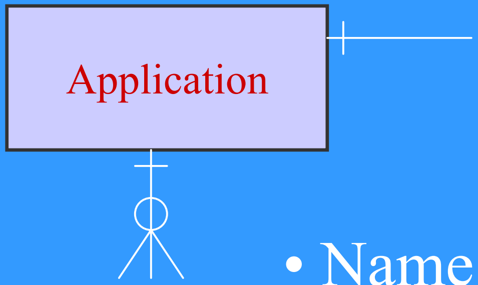
Application Security Data Elements

- Application – each application is defined to the Security Application
- User – each user is setup in the Security Application
- Role – for each application, roles are used to group users
- Program – an executable unit of an application
- Program Element – a function or data element provided by a program
- Element Value – for data elements, can be used to limit authorization levels

Expansion of Security Data Elements

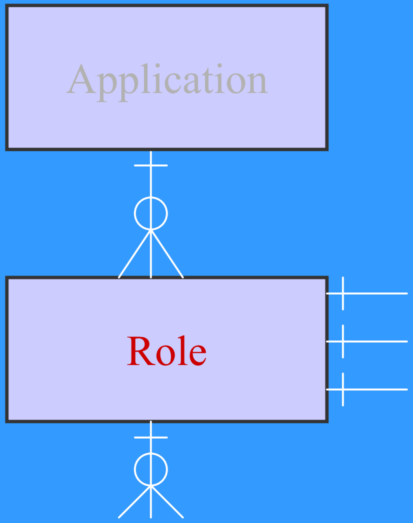


Application



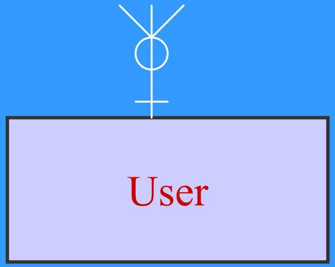
- Name
- Description
- Privilege
  - All – application is available for all users
  - None – application is not available
  - Restricted – restricted access

Role



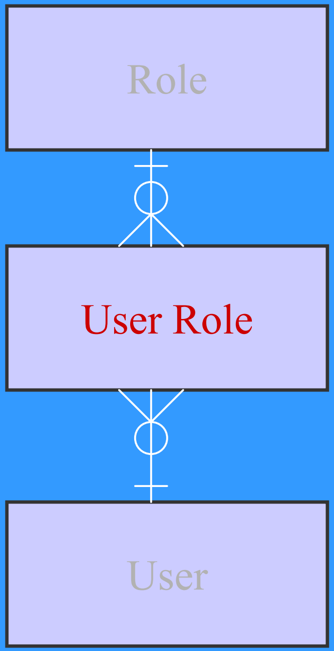
- Name
- Description
- Privilege
    - All – all application functions
    - None – no privileges
    - Restricted – restricted privileges
- Beginning Date – date privileges start
- Expiration Date – date privileges end

User



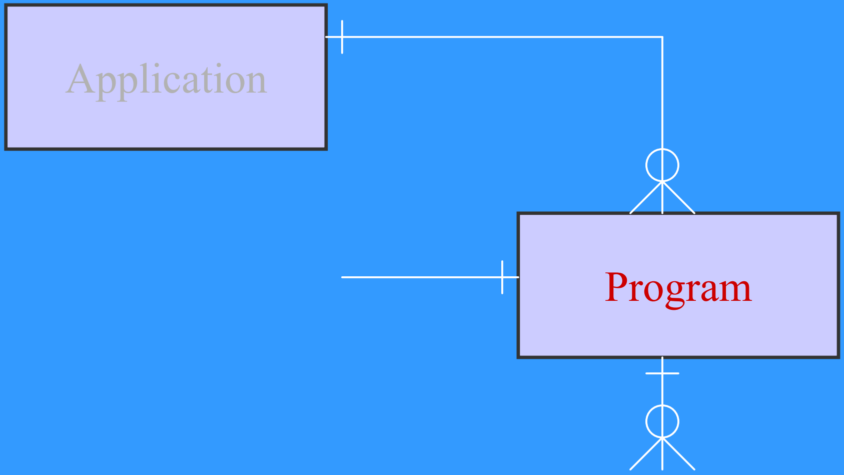
- Id
- Last Name
- First Name
- Middle Initial
- Password
- Password Expiration Date

Assignment of Users to Roles



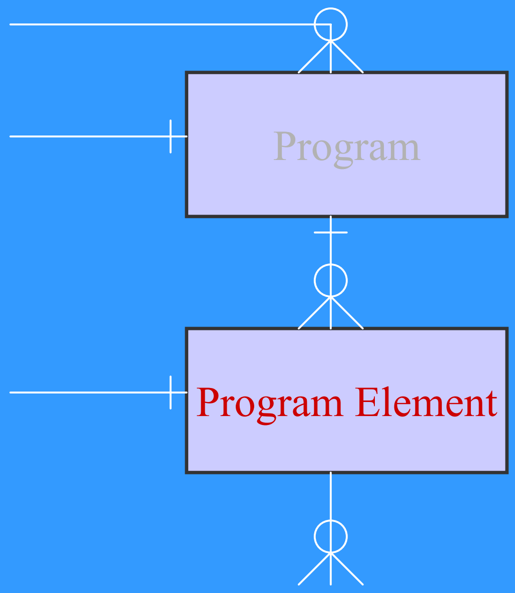
- Beginning Date
- Ending Date

Program



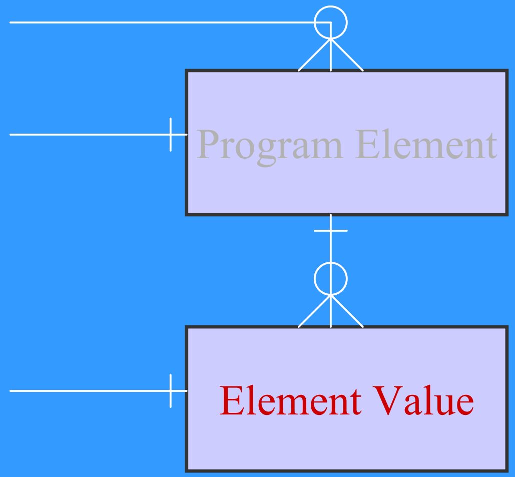
- Name
- Description
- Privilege
  - All
  - None
  - Restricted

Program Element



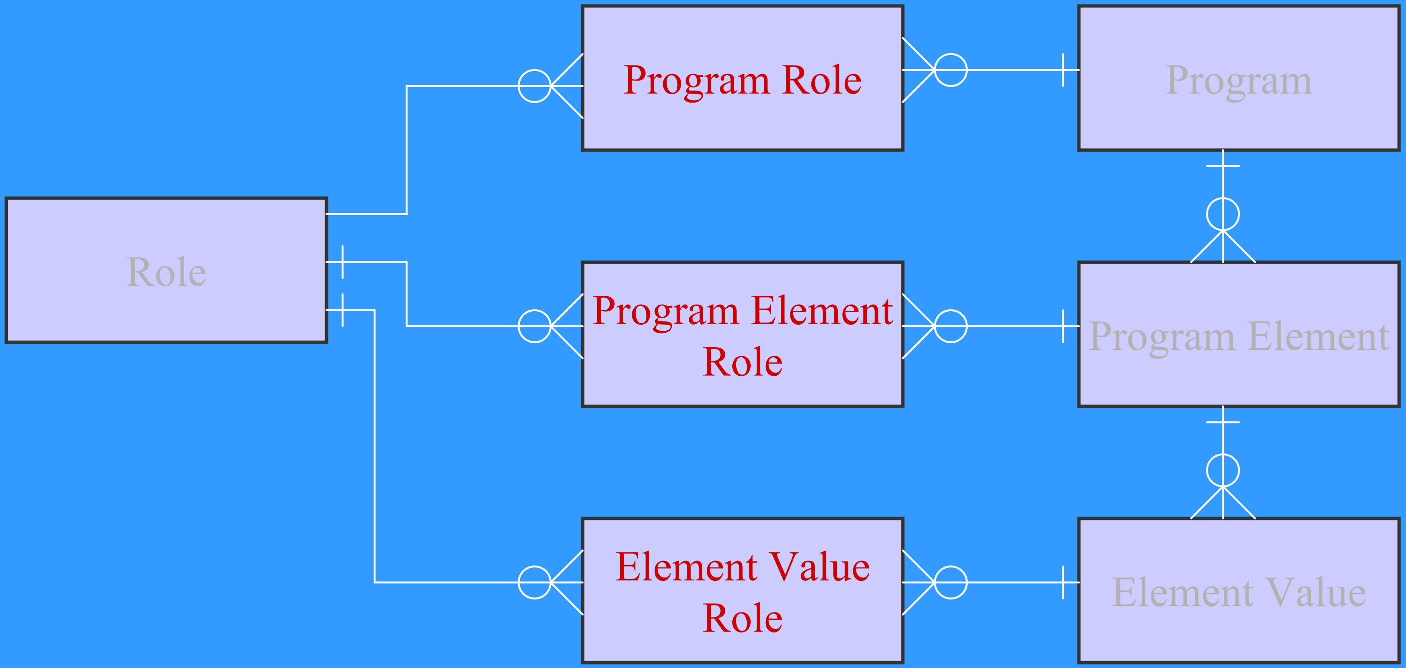
- Name
- Description
- Type
    - Control
    - Data Field

Program Element Value



- Field Value
- Field Domain

Authorization of Roles for Software



- Beginning Date
- Ending Date

Security Administration (Development)

- For each application, must decide what programs (clients) and their elements are to be secured, including value-based security
- Each client must invoke the security verification API
- Each client must interpret the results and modify its presentation layer (within its capabilities)
- Helpful to develop and use matrices to record requirements
- Adds an additional level of testing (might want to wait until core development and testing is complete)

Security Administration (Implementation)

- For each application, designate a security administrator
- Create a form and a process for requesting authorization to applications, programs, elements, etc.
- Establish a process for troubleshooting authorization problems
- Administrative effort increases with the number of roles
- Administrative effort increases with the complexity of the security requirements
- Administrative effort increases with the complexity of the interfaces (more elements to secure)

Applicability for Multiple Clients

- Users see "same" behavior across multiple clients providing the same functions
- Security services provide same results, independent of the client type or communications mechanism
- Each client type might react differently to the results, depending on the capabilities of the technology (dynamic versus static)

Possible Client Behaviors

COOL:Gen Clients

- message box presentation
- enable/disable controls

Dynamic Web Pages

- dynamic content
- provide/remove controls

Visual Basic Clients

- message box presentation
- enable/disable controls

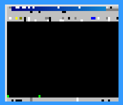
Block Mode Screens

- error message capability
- enable/disable fields

Behavior standards are really up to you…but enforce them.

COOL:Gen Client

- Require logon dialog box at application level
- Request security verification when window opens
- Request verification for all window controls
- Enable/Disable controls accordingly
- Use Message Box for authorization failure



COOL:Gen Block Mode

- Require logon screen at application level
- Request security verification on “first time” execution
- Request verification for all screen functions
- Use variables to simulate function key availability
- Or, show all functions and use message line
- Use “Authorization Failure” screen

Visual Basic Client

- Require logon dialog box at application level
- Request security verification when window opens
- Request verification for all window controls
- Enable/Disable controls accordingly
- Use Message Box for authorization failure

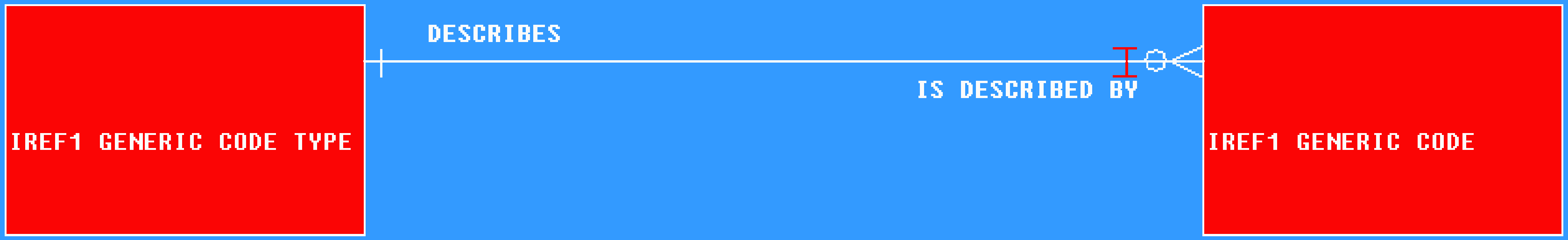
Web-Based Client

- Require logon dialog box at application level
- Request security verification when page is requested
- Request verification for all page functions
- Dynamically display controls and links accordingly
- Use Security Error page for authorization failure

Demonstration

- Review of the Security Maintenance Application web interface
- Setup a new user and authorize for access to application
- Review the actions of different client types, based on the user's authorizations

Demonstration Scenario



The demonstration application is supported by data and services of a simple “Reference” component…

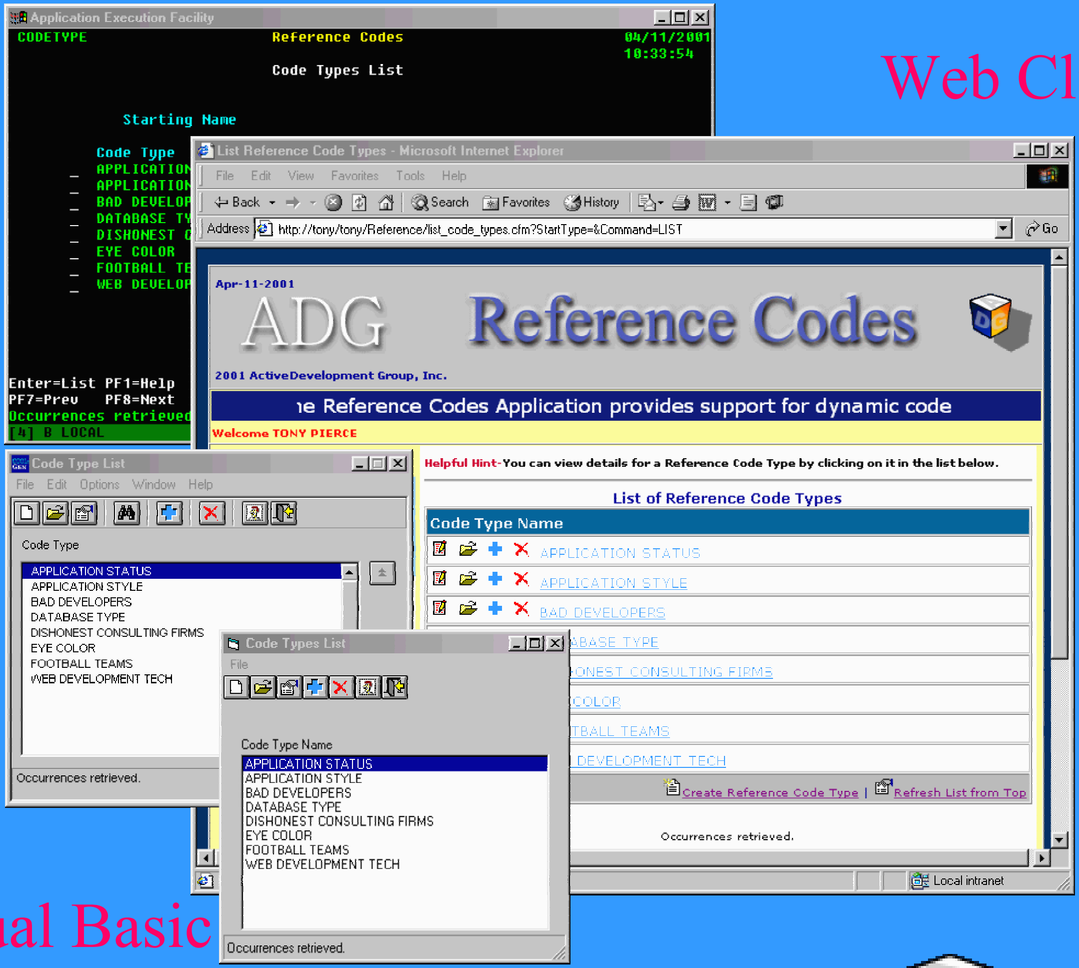
Demonstration Scenario

Block Mode

Web Client

COOL:Gen

Visual Basic


Application Execution Facility
List Reference Code Types - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address
Go
Local intranet
Code Type List
File Edit Options Window Help
Code Types List
File

```
CODETYPE                    Reference Codes            04/11/2001
                                                       10:33:54
                            Code Types List


                Starting Name

            Code Type
          _ APPLICATION
          _ APPLICATION
          _ BAD DEVELOP
          _ DATABASE TY
          _ DISHONEST C
          _ EYE COLOR
          _ FOOTBALL TE
          _ WEB DEVELOP


Enter=List PF1=Help
PF7=Prev   PF8=Next
Occurrences retrieved
[4] B LOCAL
```

http://tony/tony/Reference/list_code_types.cfm?StartType=&Command=LIST

Apr-11-2001

ADG Reference Codes

2001 ActiveDevelopment Group, Inc.

ıe Reference Codes Application provides support for dynamic code

Welcome TONY PIERCE

Helpful Hint-You can view details for a Reference Code Type by clicking on it in the list below.

List of Reference Code Types

Code Type Name
APPLICATION STATUS
APPLICATION STYLE
BAD DEVELOPERS
ABASE TYPE
ONEST CONSULTING FIRMS
COLOR
TBALL TEAMS
DEVELOPMENT TECH

Create Reference Code Type | Refresh List from Top

Occurrences retrieved.

Code Type

APPLICATION STATUS
APPLICATION STYLE
BAD DEVELOPERS
DATABASE TYPE
DISHONEST CONSULTING FIRMS
EYE COLOR
FOOTBALL TEAMS
WEB DEVELOPMENT TECH

Occurrences retrieved.

Code Type Name

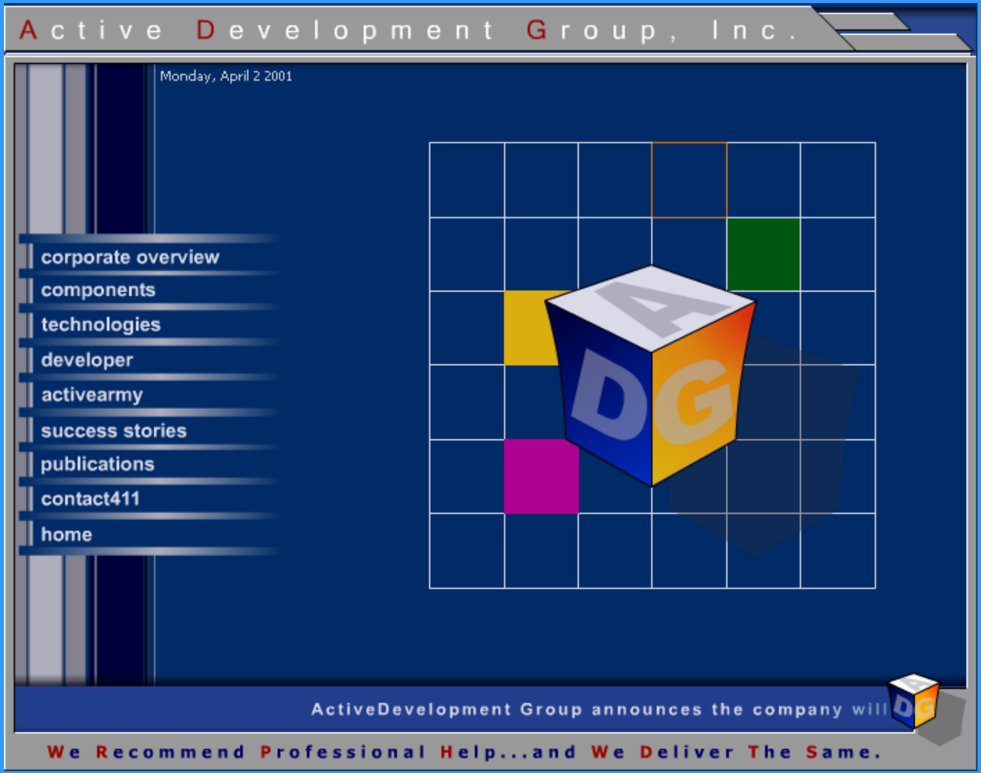
APPLICATION STATUS
APPLICATION STYLE
BAD DEVELOPERS
DATABASE TYPE
DISHONEST CONSULTING FIRMS
EYE COLOR
FOOTBALL TEAMS
WEB DEVELOPMENT TECH

Occurrences retrieved.

The demonstration includes a variety of user interfaces.

EDGE
Worldwide Community for COOL Application Developers

Questions & Comments



Active Development Group, Inc.

Monday, April 2 2001

ActiveDevelopment Group announces the company will

We Recommend Professional Help...and We Deliver The Same.

For more detailed inquires, contact techguys@ActiveDevGroup.com