

A Software Platform for Collaborative Development of Nonlinear Dynamic Analysis Code

Kincho H. Law and Jun Peng
Stanford University

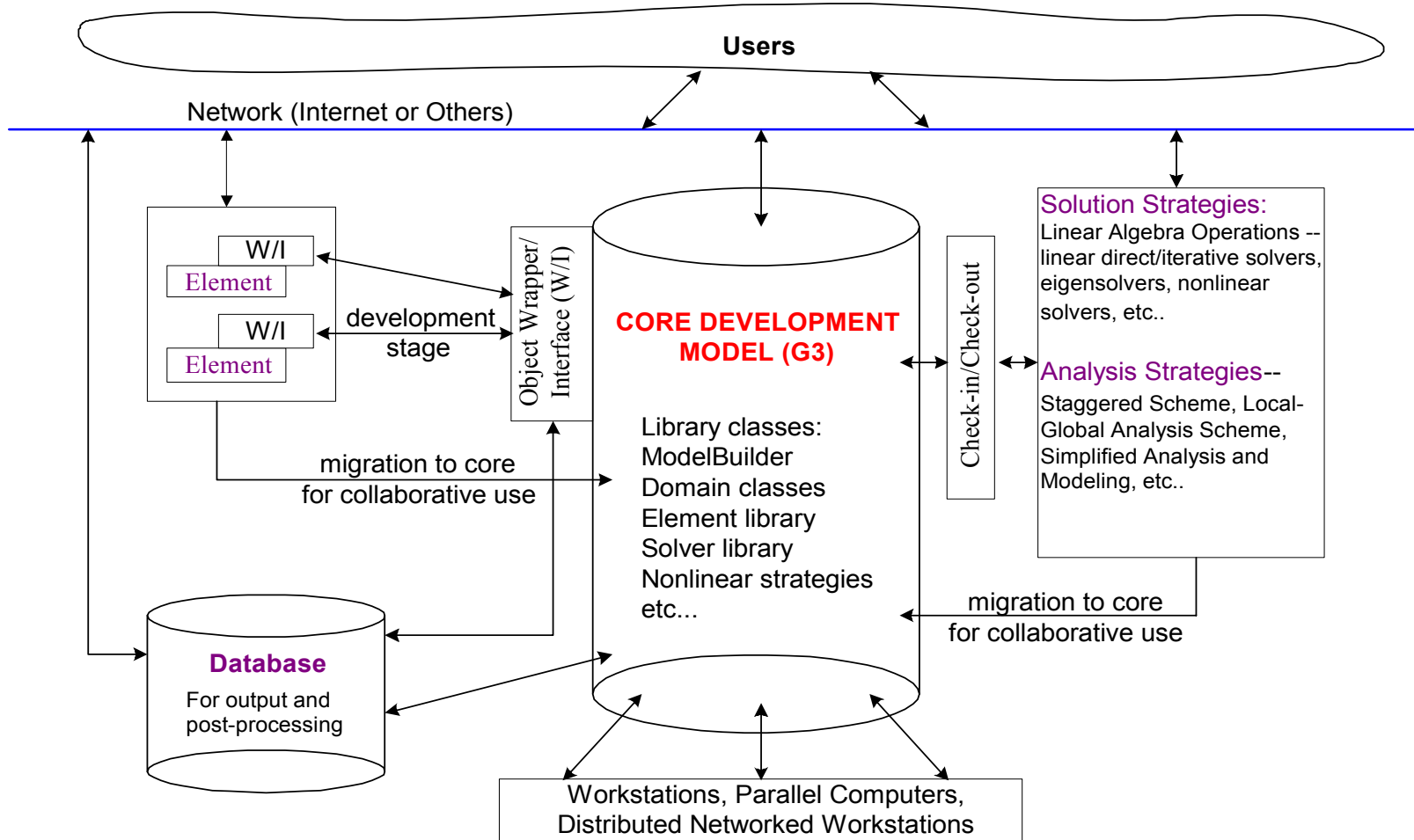
Contributions (year 2):

- Internet-based Collaborative Model
- Web Server for Source Code Release/Submission
- Distributed Element Service

Contributions (year 1):

- Integrating G3 with Symmetric-Sparse Solver
- Integrating G3 with Eigensolver (ARPACK)
- Understanding G3 Structure and Introducing New Element

G3 Collaborative Model



G3 Web Server

<http://eig.stanford.edu/g3/>

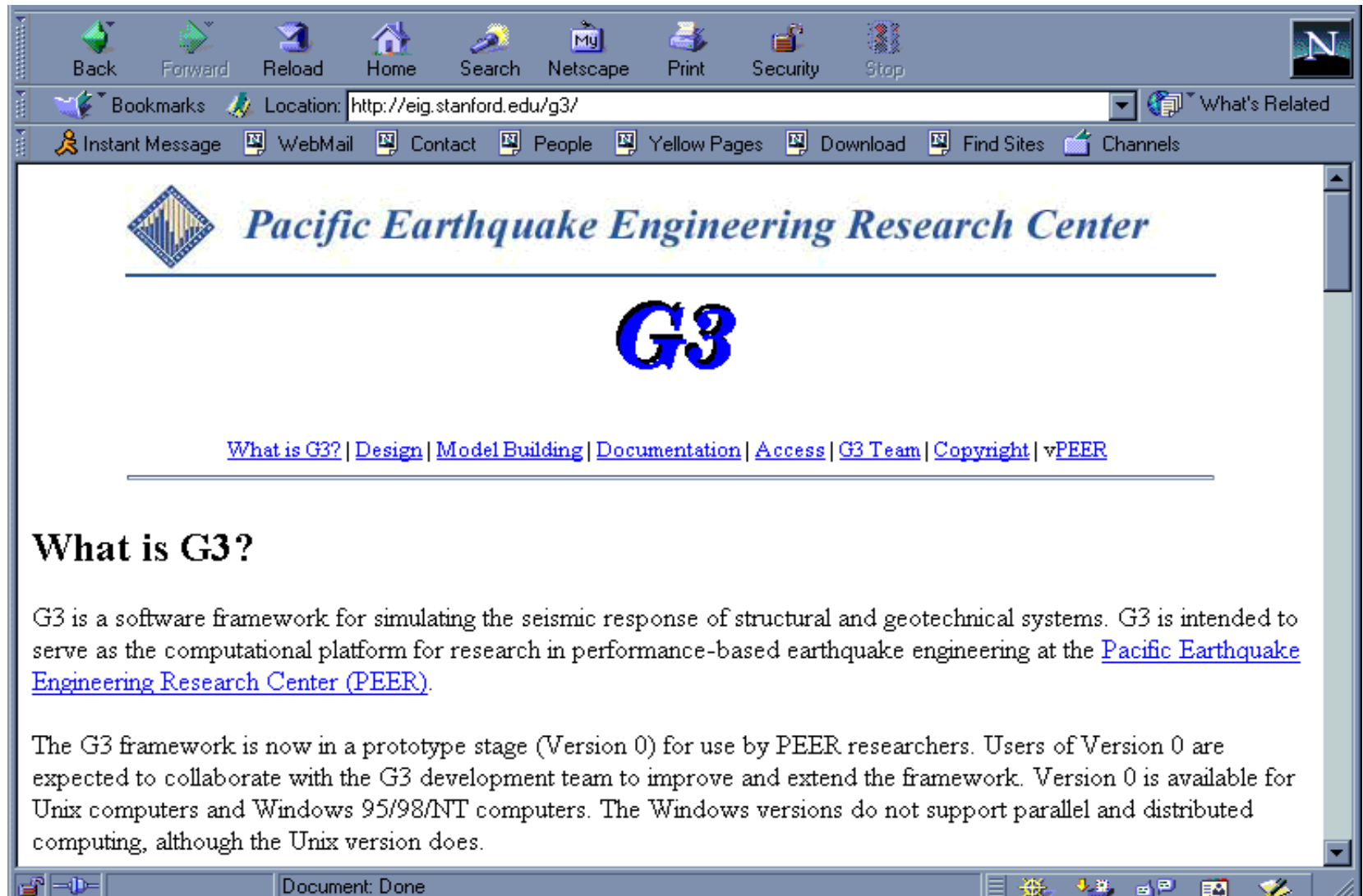
Features:

- Web Server is written in Java Servlet
- Security and Version Control are implemented in Java

Benefits:

- Remote Access: using http protocol to download and upload files
- Source Code Consistency: only one central source
- Security: access to the source code is protected by password
- Version Control: the server keeps a log of all the user's file version

G3 Web-page: <http://eig.stanford.edu/g3/>



The screenshot shows a Netscape browser window with the address bar set to <http://eig.stanford.edu/g3/>. The page content includes the logo of the Pacific Earthquake Engineering Research Center (PEER), the text "Pacific Earthquake Engineering Research Center", and a large blue "G3" logo. Below the logo is a navigation menu with links: [What is G3?](#), [Design](#), [Model Building](#), [Documentation](#), [Access](#), [G3 Team](#), [Copyright](#), and [PEER](#). The main heading "What is G3?" is followed by two paragraphs of text describing the G3 software framework.

Pacific Earthquake Engineering Research Center

G3

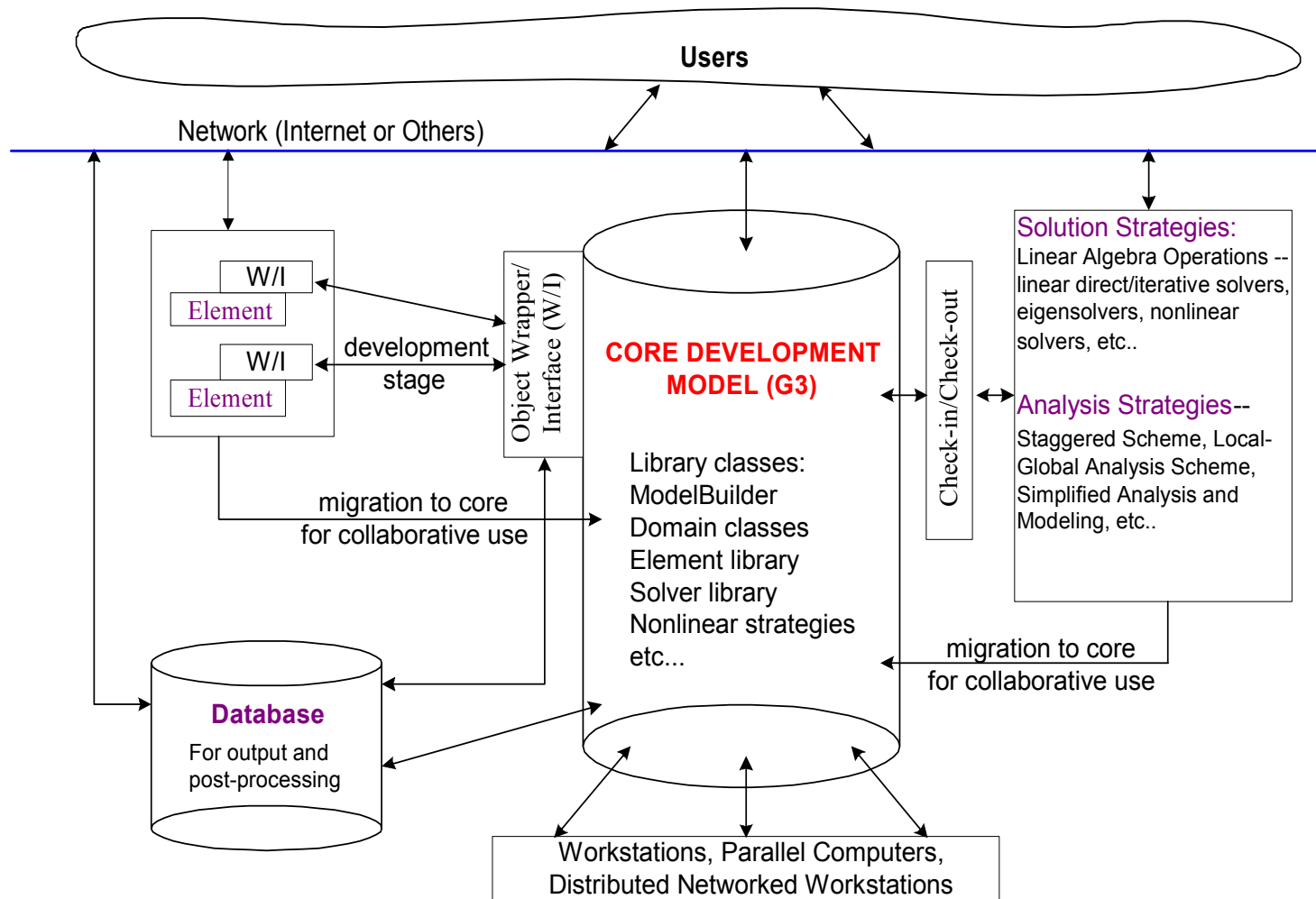
[What is G3?](#) | [Design](#) | [Model Building](#) | [Documentation](#) | [Access](#) | [G3 Team](#) | [Copyright](#) | [PEER](#)

What is G3?

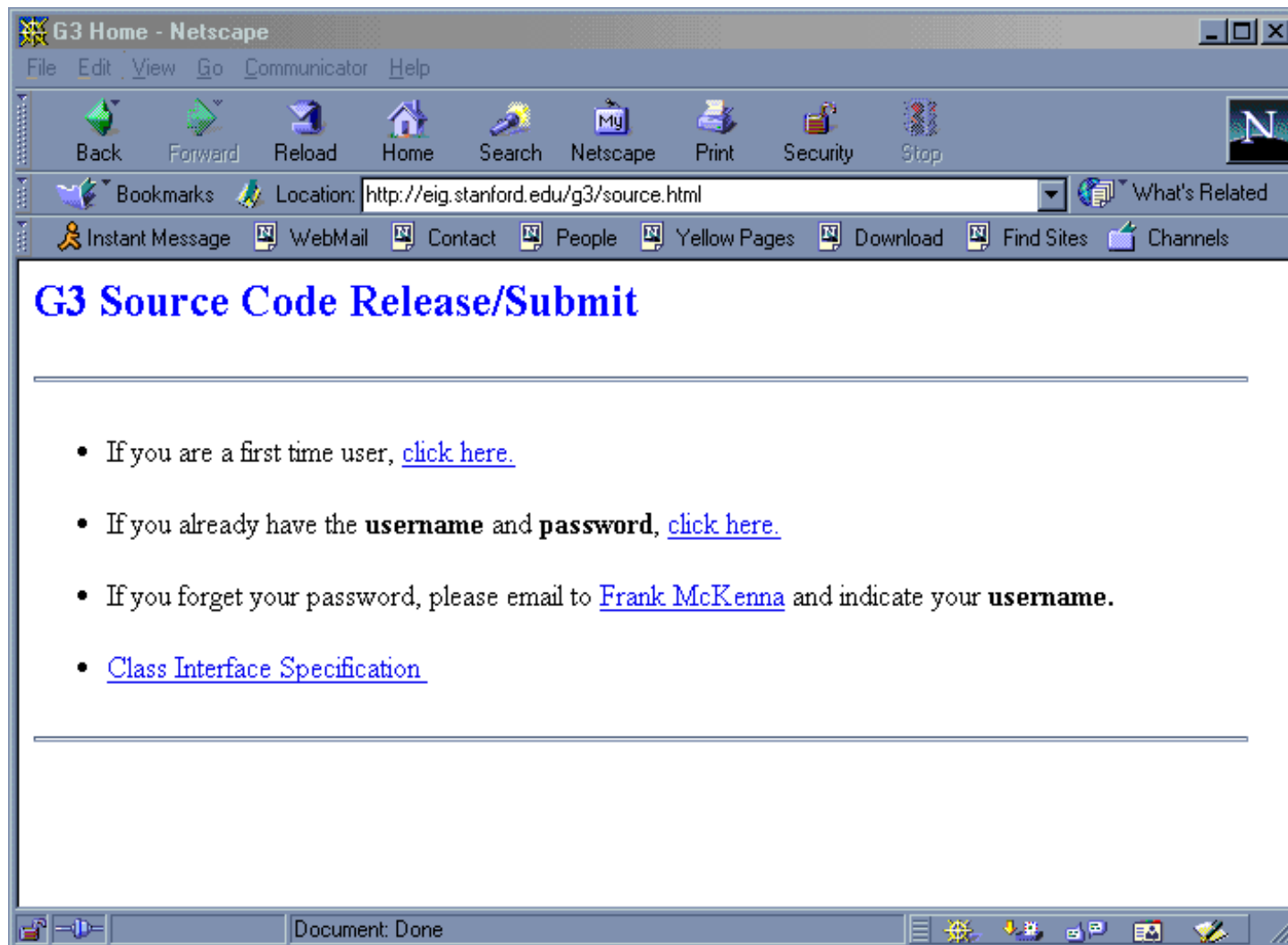
G3 is a software framework for simulating the seismic response of structural and geotechnical systems. G3 is intended to serve as the computational platform for research in performance-based earthquake engineering at the [Pacific Earthquake Engineering Research Center \(PEER\)](#).

The G3 framework is now in a prototype stage (Version 0) for use by PEER researchers. Users of Version 0 are expected to collaborate with the G3 development team to improve and extend the framework. Version 0 is available for Unix computers and Windows 95/98/NT computers. The Windows versions do not support parallel and distributed computing, although the Unix version does.

G3 Web Server



G3 Web Server: Source Code Release/Submit



G3 Web Server: Registration Form

G3 Registration Form

Please complete the following form. After you submit this form, an email will be sent to you to confirm the registration in about two days.

You need to provide all the information required below to complete the registration. Only the comment field is optional.

[Click here](#) to return to G3 Homepage.

Please enter the Username and your preferred Password:

Username:

Password:

Re-enter Password:

Please enter some of your personal information:

Last Name:

First Name:

E-mail address:

University:

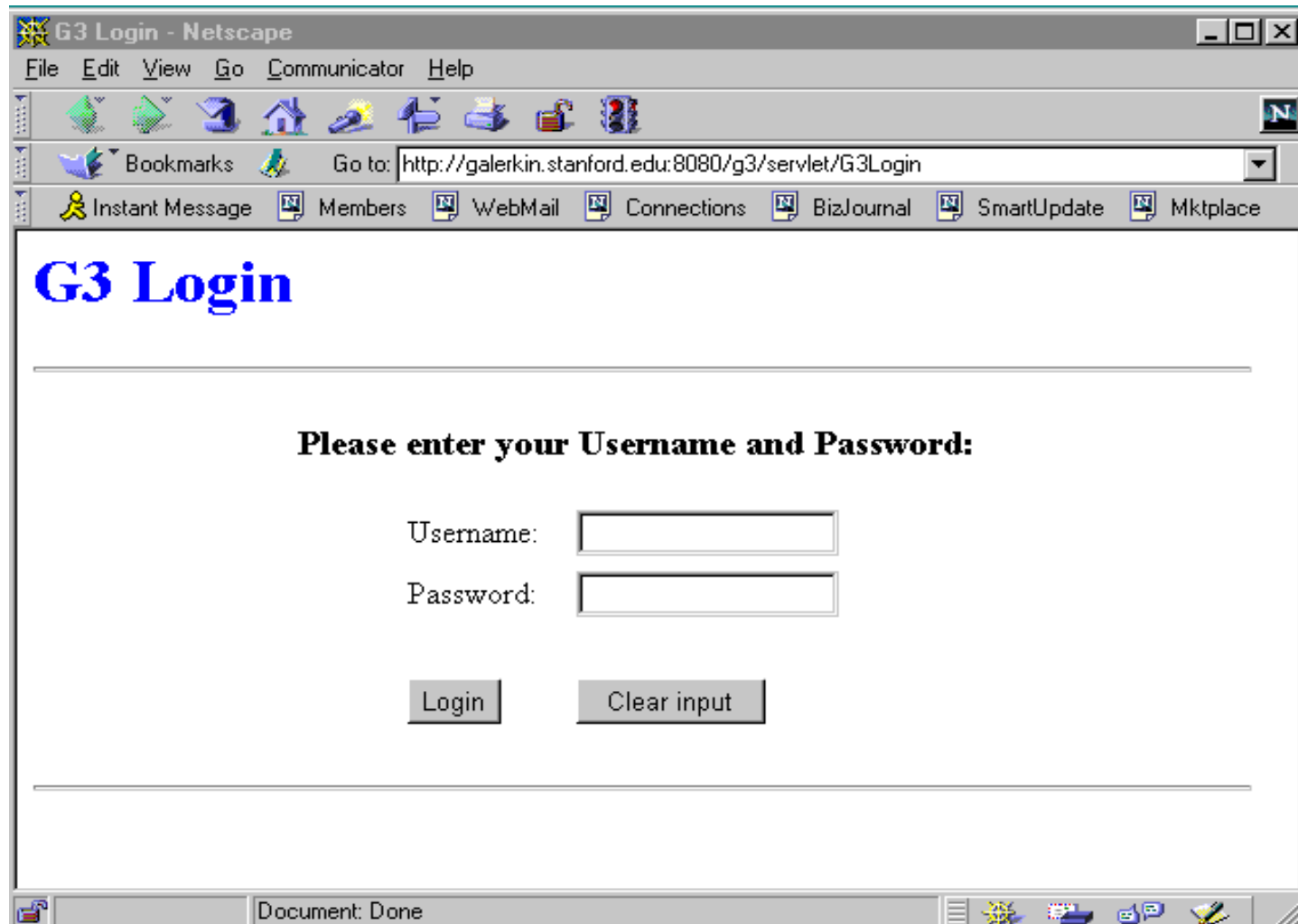
Prof.(advisor) Name:



Document: Done



G3 Web Server: Login



G3 Web Server: Front Page

The screenshot shows a web browser window with a title bar. The page content includes a main heading, a navigation menu, a comment input area, and submission buttons.

G3 Front Page

Front Page List All Files Updated Files Downloaded Files Submit File Logout

Please enter comments and suggestions:

A large, empty text input area with a vertical scrollbar on the right side and horizontal scrollbars at the bottom.

Submit Comment Clear Input

G3 Web Server: List of All the Files

List of All the Files

[Front Page](#)[List All Files](#)[Updated Files](#)[Downloaded Files](#)[Submit File](#)[Logout](#)

In order to download a file, just click the corresponding **Download** button. This will bring your local file system for you to choose which directory you want the file to be saved.

For windows files, please use **winzip** to do unzip. For unix files, please use "**gunzip filename.tar.gz**" followed by "**tar xvf filename.tar**". For further instructions, please read the **README** file.

FileName	Operating System	Download
g3r0.zip	Windows 95/98/NT	Download
g3r0.tar.gz	Unix	Download

Please always Logout when you are done: [Logout](#)

G3 Web Server: Download File

List of All Files

Front Page | [Logout](#) | [File](#) | [Logout](#)

In order to download files, you need to choose which directory you want to download to on your local file system for you.

For windows files, please use "filename.tar.gz" followed by "tar" to indicate the file format.

Save As...

Save in: g3

File name: g3r0

Save as type: All Files (*.*)

Save Cancel

FileName	Operating System	Download
g3r0.zip	Windows 95/98/NT	Download
g3r0.tar.gz	Unix	Download

Please always Logout when you are done: [Logout](#)

G3 Web Server: List of the Updated Files

List of the Updated Files

[Front Page](#)[List All Files](#)[Updated Files](#)[Downloaded Files](#)[Submit File](#)[Logout](#)

Here are the files that have been updated since the last time you logged in. You can download the file(s) to update your local copy of G3. After you done, please click **Logout** button, otherwise, your log file will not be updated!

FileName	Operating System	Download
g3r0.tar.gz	Unix	Download

Please always Logout when you are done: [Logout](#)

G3 Web Server: File Submission

G3 File Submission

[Front Page](#)[List All Files](#)[Updated Files](#)[Downloaded Files](#)[Submit File](#)[Logout](#)

Use the **Browse...** button to access your local file system to choose the file that you want to submit. When the file name appears in the text field, click **Submit File** button to submit the file.

To make some comments or give us the information about the file that you have submitted, please use the comment box in the Front Page by clicking the **Front Page** button.

Enter data file below:

[Browse...](#)[Submit File](#)

Please always Logout when you are done: [Logout](#)

Documentation: Class Interface (by package)

[About this document...](#)

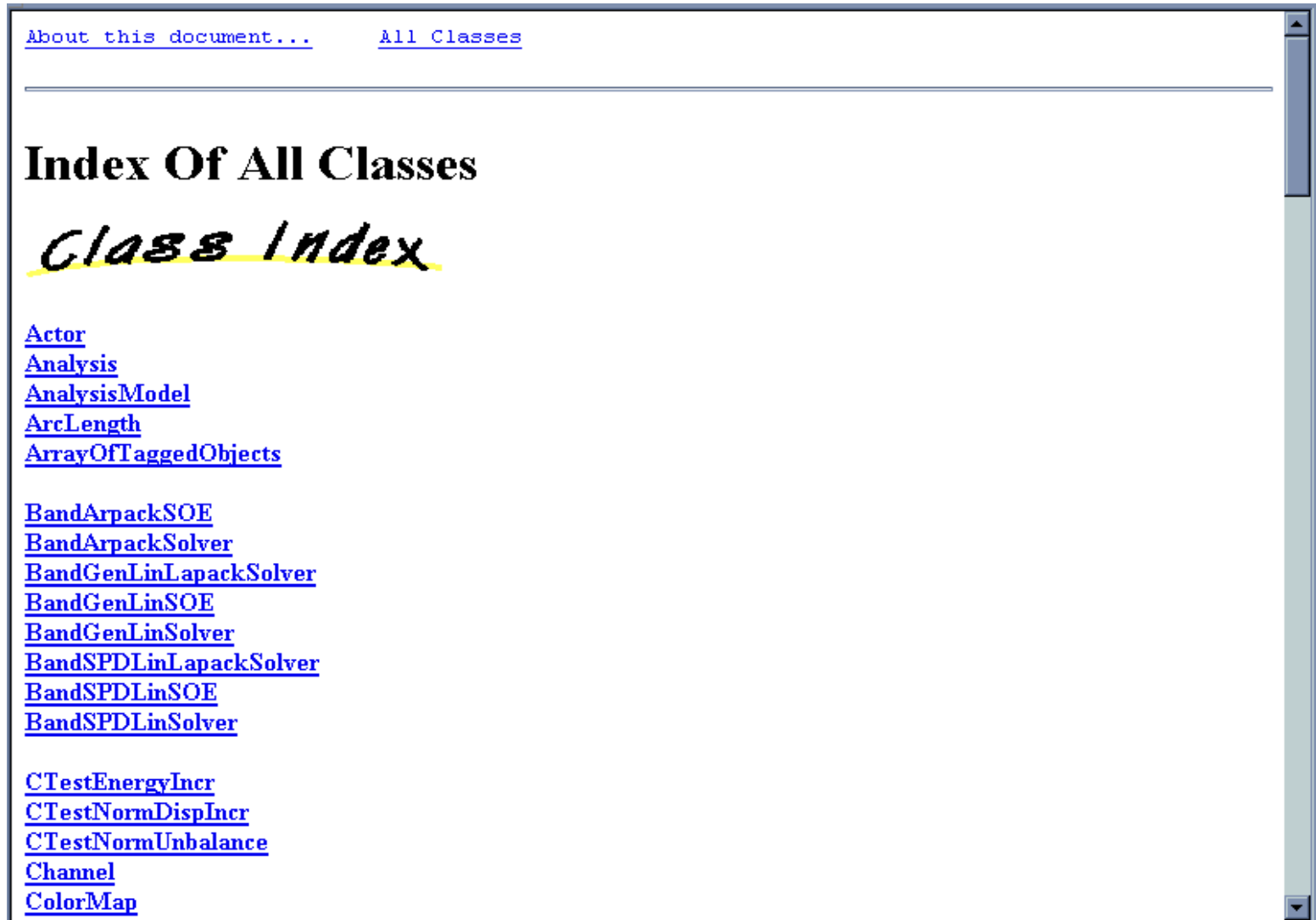
[Index](#)

G3 API Specification

Class Index

- [Matrix Classes](#)
 - [Matrix](#)
 - [Vector](#)
 - [ID](#)
- [Domain Classes](#)
 - [DomainComponent](#)
 - [Element](#)
 - [Node](#)
 - [Load](#)
 - [NodalLoad](#)
 - [ElementalLoad](#)
 - [EarthquakeLoad](#)
 - [SingleExcitation](#)
 - [SP Constraint](#)
 - [MP Constraint](#)
 - [Domain](#)
- [Analysis Classes](#)
 - [Analysis](#)
 - [StaticAnalysis](#)
 - [TransientAnalysis](#)

Documentation: Class Interface (by Index)



[About this document...](#) [All Classes](#)

Index Of All Classes

Class Index

[Actor](#)
[Analysis](#)
[AnalysisModel](#)
[ArcLength](#)
[ArrayOfTaggedObjects](#)

[BandArpackSOE](#)
[BandArpackSolver](#)
[BandGenLinLapackSolver](#)
[BandGenLinSOE](#)
[BandGenLinSolver](#)
[BandSPDLinLapackSolver](#)
[BandSPDLinSOE](#)
[BandSPDLinSolver](#)

[CTestEnergyIncr](#)
[CTestNormDispIncr](#)
[CTestNormUnbalance](#)
[Channel](#)
[ColorMap](#)

Documentation: Class Element Interface (1)

[All Classes](#) [This Package](#) [Previous](#) [Next](#) [Index](#)

Class Element

[TaggedObject](#)

```
graph TD
    TaggedObject --|> MovableObject
    MovableObject --|> DomainComponent
    DomainComponent --|> Element
```

Files:
#include <~/element/Element.h>

Class Declaration:
class Element: public [DomainComponent](#)

Description:
Element is an abstract class, i.e. no instances of Element will exist. The element class provides the interface that all element writers must provide when introducing new element classes.

Constructor Index

[Element\(int tag, int classTag\):](#)

Documentation: Class Element Interface (2)

• virtual Matrix &getMass

virtual [Matrix](#) &getMass (void) =0;

To return the mass matrix. The element is to compute its mass matrix based on the original location of the nodes and the current trial response quantities at the nodes.

$$\mathbf{M}_e = \frac{\partial \mathbf{F}_{I_i}}{\partial \ddot{\mathbf{U}}}|_{\mathbf{U}_{trial}}$$

• virtual void zeroLoad

virtual void zeroLoad (void) =0;

This is a method invoked to zero the element load contributions to the residual, i.e. $\mathbf{P}_e = \mathbf{0}$

• virtual Vector &getResistingForce

virtual [Vector](#) &getResistingForce (void) =0;

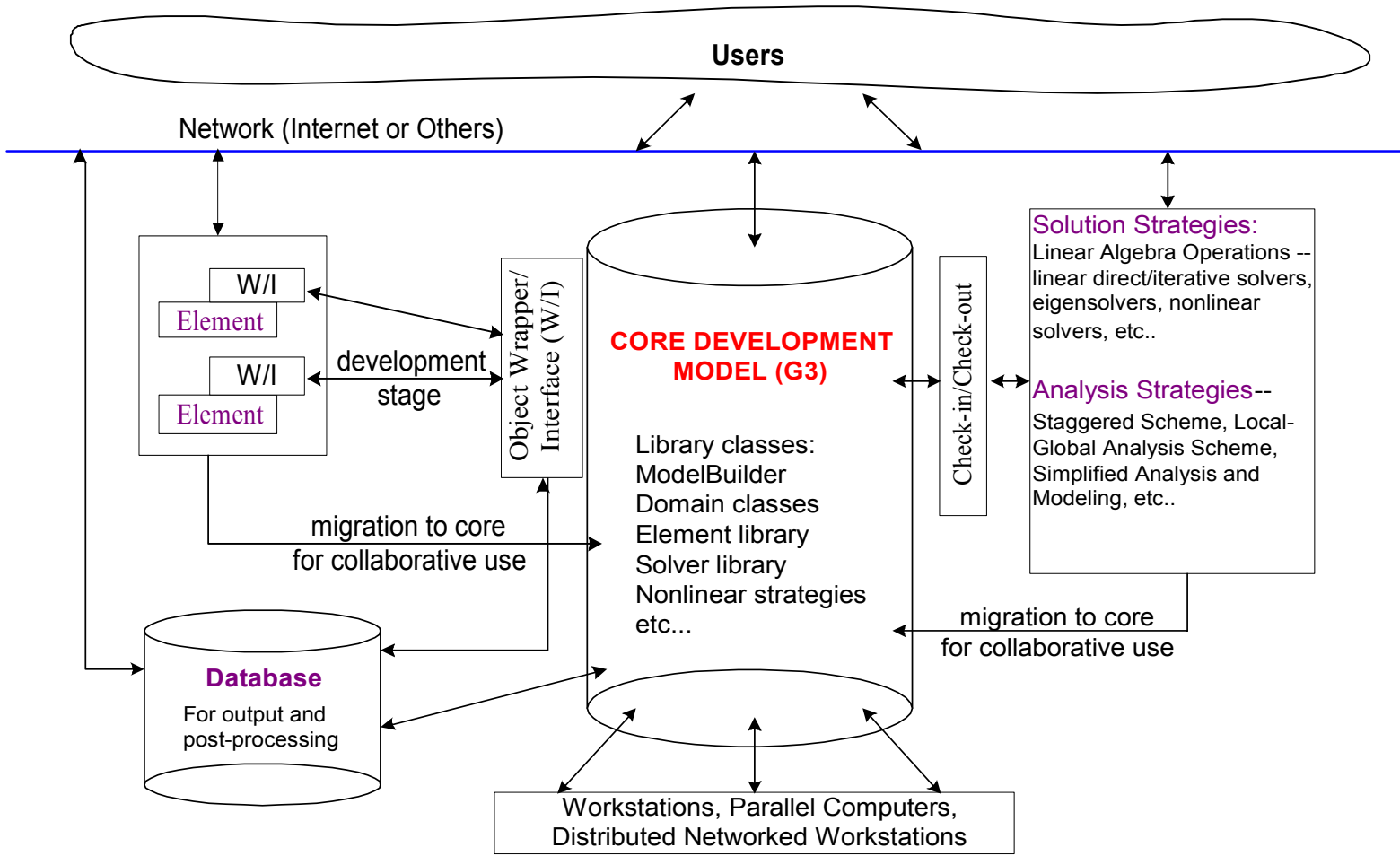
Returns the resisting force vector for the element. This is equal to the applied load due to element loads minus the loads at the nodes due to internal stresses in the element due to the current trial displacement, i.e.

$$\mathbf{R}_e = \mathbf{P}_e - \mathbf{F}_{R_e}(\mathbf{U}_{trial})$$

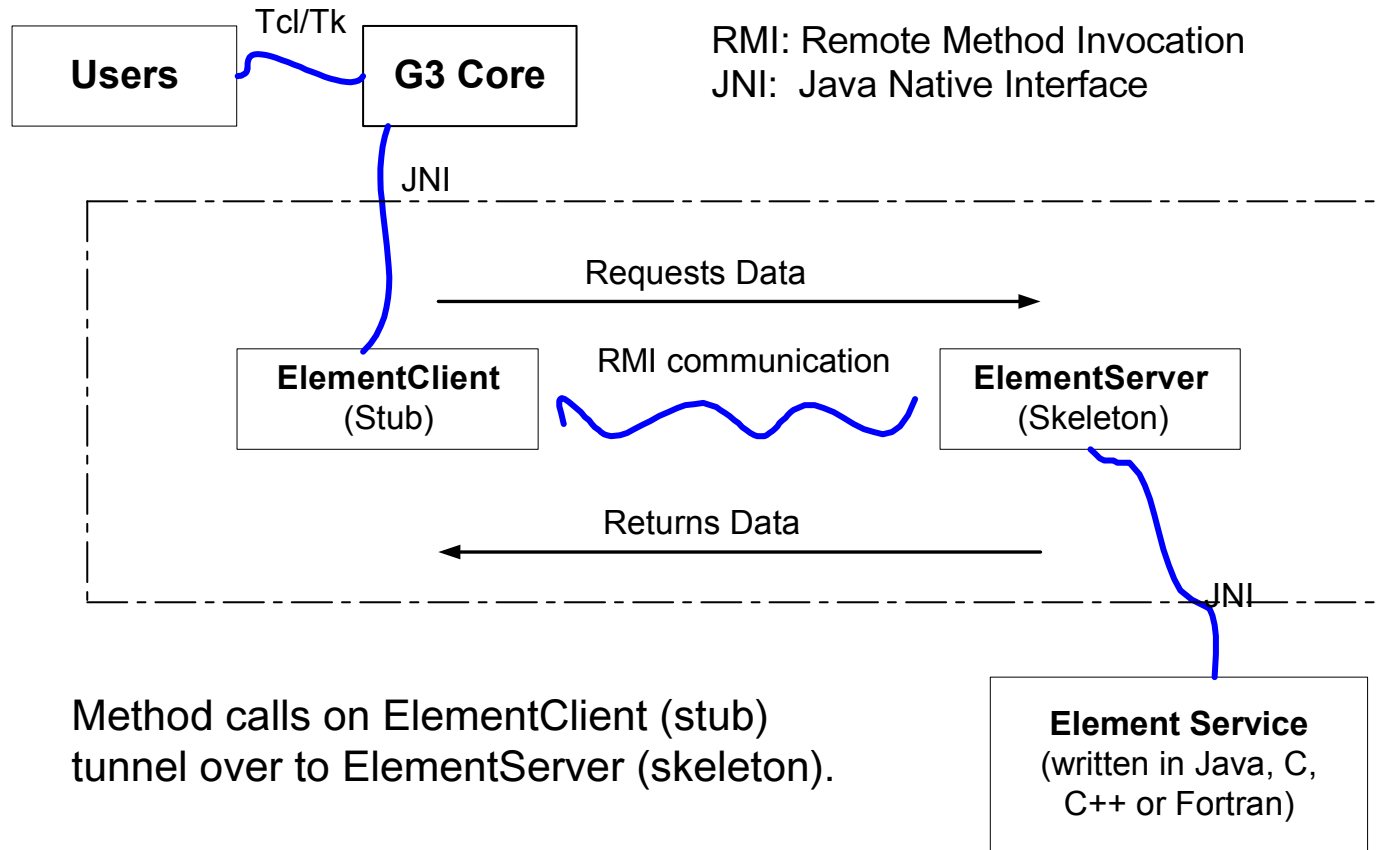
• virtual Vector &getResistingForceIncInertia

virtual [Vector](#) &getResistingForceIncInertia (void) =0;

Distributed Element Service for G3



Distributed Element Service



Distributed Element Service: Class Interface

```
public class ElementRemote extends Remote {
    // This is the service name for publishing.
    public static final String SERVICE = "ElementService";
    // This is the port number, could be changed as needed.
    public static final int PORT = 1234;

    // This function is used to send the element data to server.
    public void formElement(int tag, Identity src, String input);
    // When the analysis finished, use this function to do housecleaning.
    public void clearElements(Identity src);

    public int commitState(int tag, Identity src);
    public int revertToLastCommit(int tag, Identity src);
    public int revertToStart(int tag, Identity src);

    // Form element stiffness, damping and mass matrix.
    public MyMatrix getTangentStiff(int tag, Identity src);
    public MyMatrix getSecantStiff(int tag, Identity src);
    public MyMatrix getDamp(int tag, Identity src);
    public MyMatrix getMass(int tag, Identity src);

    public void zeroLoad(int tag, Identity src);
    public MyVector getResistingForce(int tag, Identity src);
    public MyVector getTestingForceIncInertia(int tag, Identity src);
}
```

Distributed Element Service: Interaction Diagram

