UTNP and Model Integration Method
Introduction of Example Structures

Xu Huang

Civil Engineering
UNIVERSITY OF TORONTO

Outline

- **Model Integration Method**
- **Communication method (UTNP)**
  - Communication protocol
  - Data exchange format
  - Dynamic Link Library (DLL)
- **Integration Modules**
- **Substructure Modules**
- **Example Structures**

Model Integration Method

- **Hybrid simulation is a basically substructuring technique.**

$$\boldsymbol{m\ddot{u} + c\dot{u} + r = f(t)}$$

- **General requirements in hybrid simulations**
  - Main numerical integration module
    - Dynamic problem: direct integration algorithms
    - Static problem: nonlinear solution scheme
  - Substructure module
    - Numerical elements
    - Physical specimens
  - Communication between integration and substructure modules

Model Integration Method

- **Assumption for integration**
  - Substructure is either displacement or velocity dependent

$$\boldsymbol{m\ddot{u} + c\dot{u} + r(u, \dot{u}) = f(t)}$$

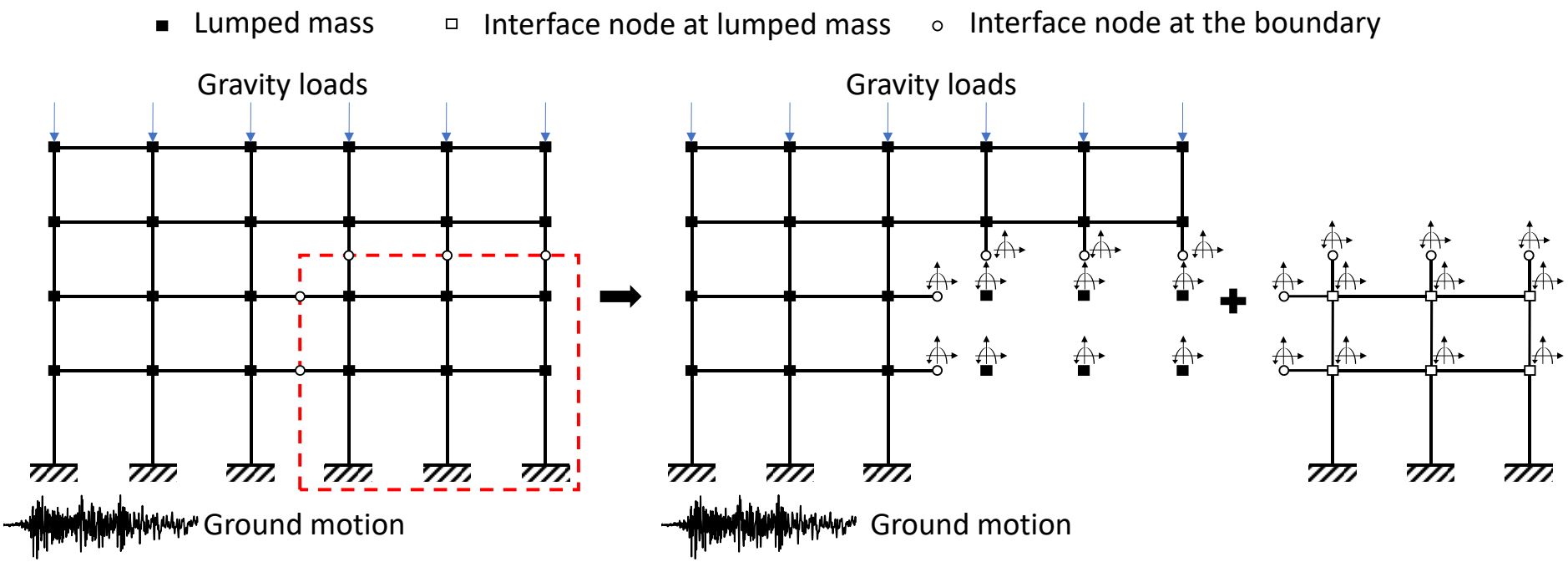
  - Substructure modules only contribute to the restoring force term, $\boldsymbol{r(u, \dot{u})}$

$$\boldsymbol{m\ddot{u} + c\dot{u} + r^{I}(u, \dot{u}) + \sum r^{S}(u, \dot{u}) = f(t)}$$

Model Integration Method

❑ Example

$$m\ddot{u} + c\dot{u} + r^{I}(u,\dot{u}) + \sum r^{S}(u,\dot{u}) = f(t)$$

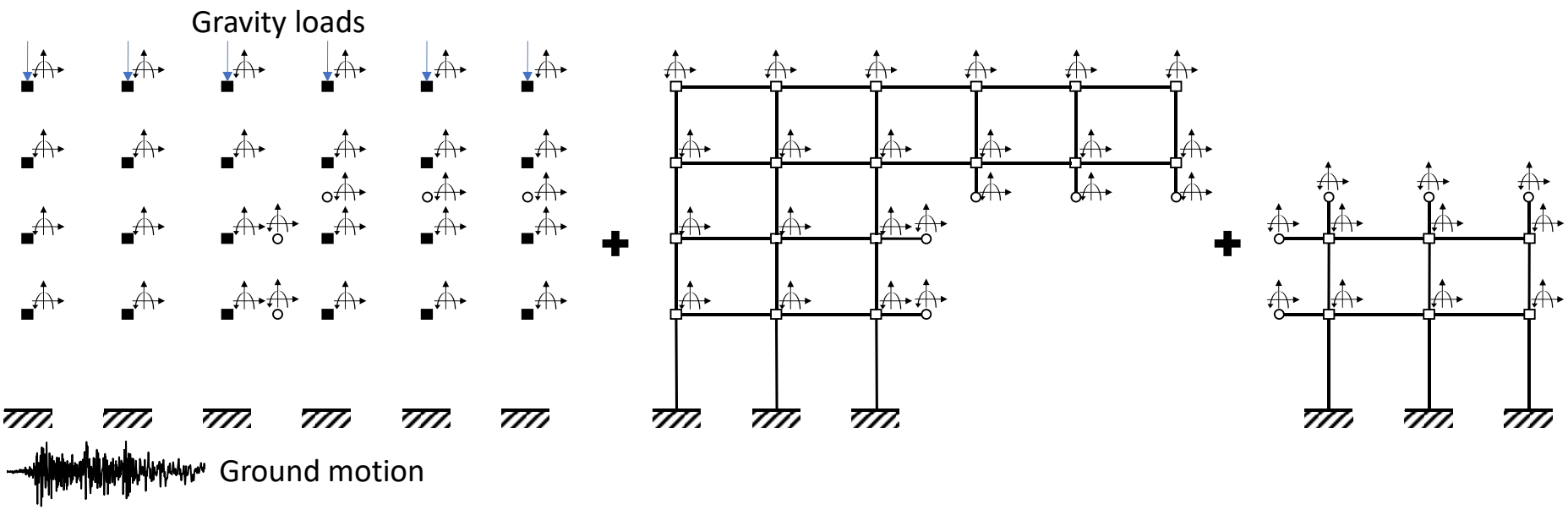


(a) Entire structure (b) Integration module (c) Substructure module

Model Integration Method

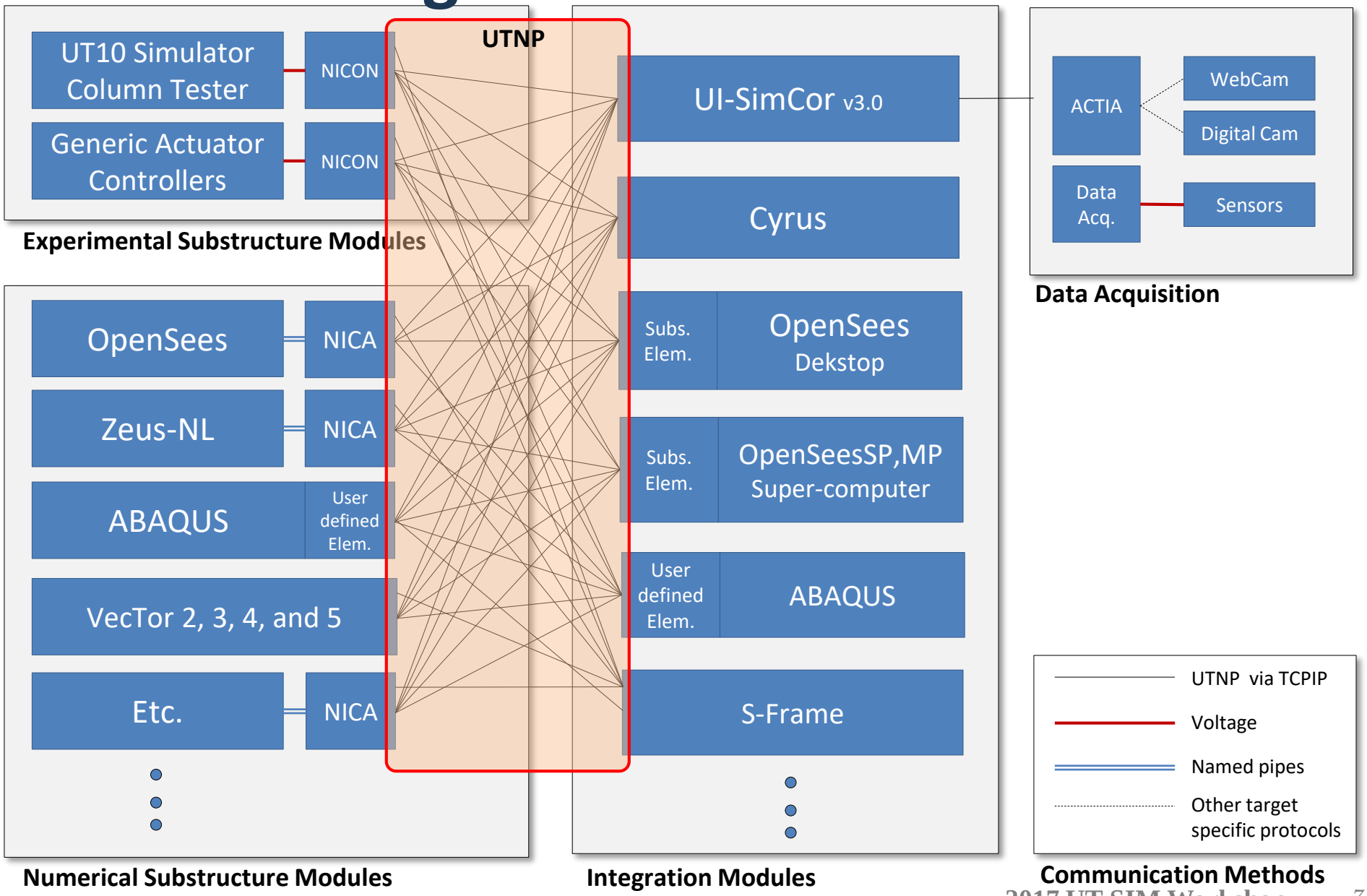
- **Example**

$$m\ddot{u} + c\dot{u} + \sum r^S(u, \dot{u}) = f(t)$$



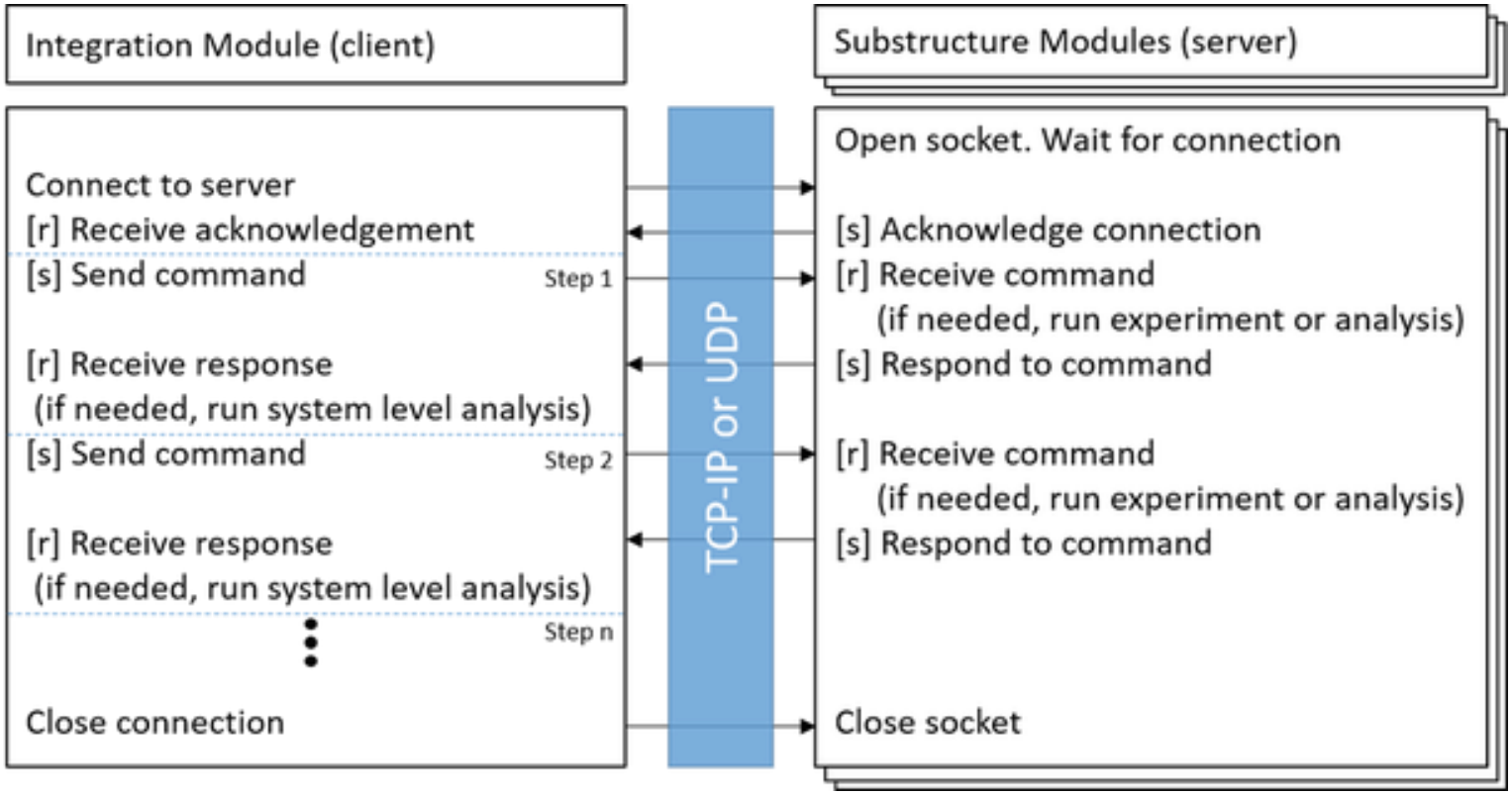
(a) Integration module (b) Substructure module 1 (c) Substructure module 2

Model Integration Method



Communication Method (UTNP)

- **Standardized communication protocol**
  - A rule that two communication modules need to follow to exchange data
  - Current development: Sequential communication



Communication Method (UTNP)

- **Standardized data exchange format**
  - Structure of data to be transmitted between an integration module and substructure modules
  - Cover parameters and indicators that are needed for various simulation purposes (i.e. hybrid thermal simulation, real-time simulation, etc.)
  - Have flexibility for future extension

32 bit

Version	Command	Test type	Substructure type
Precision	Data type	Number of DOFs	
Step number		Reserved for future extension	
Time stamp			
Data			

16 bytes

Size depends on header

Communication Method (UTNP)

❑ Parameters

Data filed	Byte	Type	Description
Version	1	uint8	UTNP version
Command	1	uint8	**3** – Impose target values specified in the Data Type filed **10** – Ask for values specified in the Data Type field **99** - Terminate communication … expandable to 256 commands in the future
Data type	1	uint8	**Bit 1** – displacement **Bit 2** – velocity **Bit 3** – acceleration **Bit 4** – restoring force **Bit 5** – stiffness matrix **Bit 6** – mass matrix **Bit 7** – temperature load **Bit 8** – future extension
Number of DOFs	2	uint16	Total number of degrees of freedoms at the interface nodes

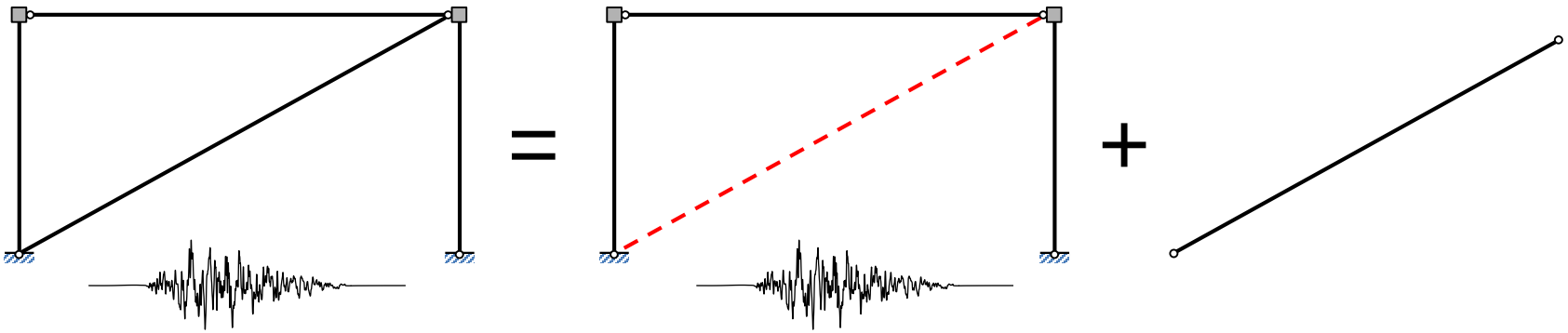
Communication Method (UTNP)

❑ Parameters (continued)

Precision	1	uint8	The precision of data appended to the header **1** – Single precision (4 bytes) **2** – Double precision (8 bytes)
Test type	1	uint8	Defines the type of simulation (for information only) **1** - Pseudo-dynamic (ramp-hold) **2** - Pseudo-dynamic (continuous) **3** - Real-time **4** - Software-only
Substructure type	1	uint8	Define the type of the substructure module used in the simulation **1** – OpenSees **2** – Zeus-NL **3** – ABAQUS **4** – VecTor programs **5** – Test equipment (hybrid simulation) … expandable to 256 substructure types
Time stamp	4	uint32	Current time stamp
Reserved for extension	2	uint16	

Communication Method (UTNP)

❑ Example



Complete model

Integration module

Substructure module (OpenSees)

Version	Command (3)	Test type (4)	Substructure type (1)
Precision (2)	Data type (10000000)	Number of DOFs (4)	
Step number (1)		Reserved for future extension	
Time stamp			

Communication Method (UTNP)

❑ Functions in the library or Object

Main Functions	Description
SetupConnection (*Port, Socket, Flag, Addr, Protocol*)	Establish a connection to a specified socket
UpdateMessageHeader (*Version, Command, TestType, SubType, Precision, Numdofs*)	Initialize the message header for communication
Initializaton (*Socket, Flag, Protocol*)	Synchronize the message headers on both client and server sides
SendData (*Socket, sData, Size, Protocol*)	Send data
RecvData (*Socket, rData, Size, Protocol*)	Receive data
UpdateDataType (*Disp, Vel, Accel, Force, Stiff, Mass, Temp*)	Update data type parameter in the message header
Indicator (*void*)	Calculate the size of data to be appended to the message header
Terminate (*Socket*)	Disconnect and close socket

Communication Method (UTNP)

- **Encapsulate data exchange format and communication related functions**
  - Shared library or shared object file (***DataExchange.dll*** or ***DataExchange.out***)
  - Compile all functions as a DLL or Object, and then use the same code in all applications
  - Designed in such a way that it is compatible with different languages
    - C++ (OpenSees, NICA)
    - Fortran (VecTor programs, ABAQUS)
    - LabVIEW (NICON)
    - MATLAB (UI-SimCor)

Integration Modules

- **Main tasks**
  - Run numerical time integration schemes
  - Main solver for static or equivalent static equation.

$$\mathbf{AX} = \boldsymbol{B}$$

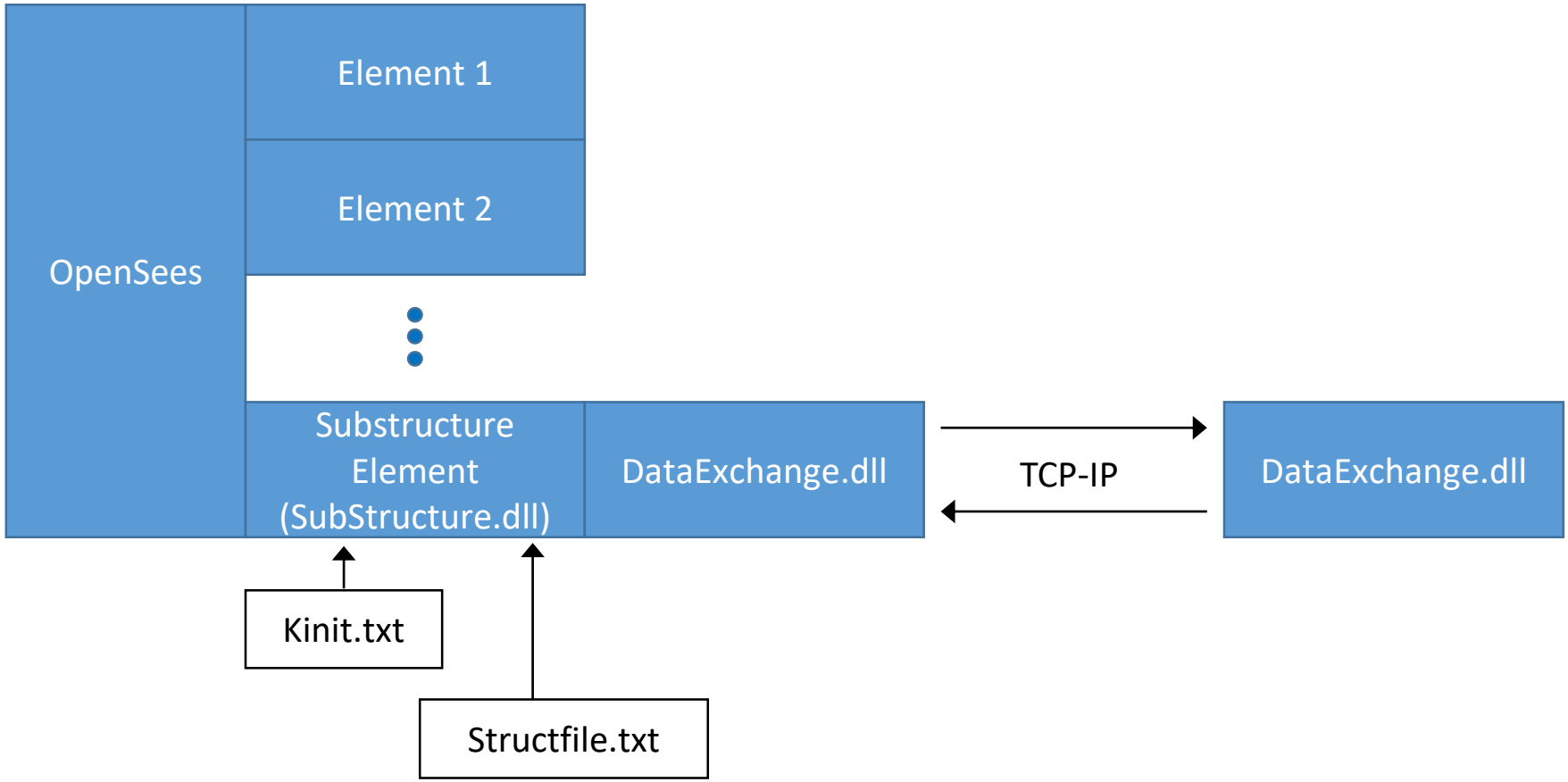
- **Requirements in integration schemes and nonlinear solution methods**
  - As few number of basic communication loops as possible for each time step calculation
  - If implicit schemes are used, displacement increments calculated by the nonlinear solution scheme for each time step should be monotonically increasing or monotonically decreasing.
  - The stiffness of the element represented in the substructure module for nonlinear iterations can be easily retrieved.

Integration Modules

- **Potential integration scheme and nonlinear solution methods**
  - Non-iterative explicit integration methods
    - i.e. Central Difference method, explicit Newmark method, explicit generalized-α method, etc.
  - Non-iterative implicit-explicit integration methods
    - i.e. α-OS method, P-C method, etc
  - Iterative implicit integration methods
    - i.e. initial stiffness for nonlinear iteration

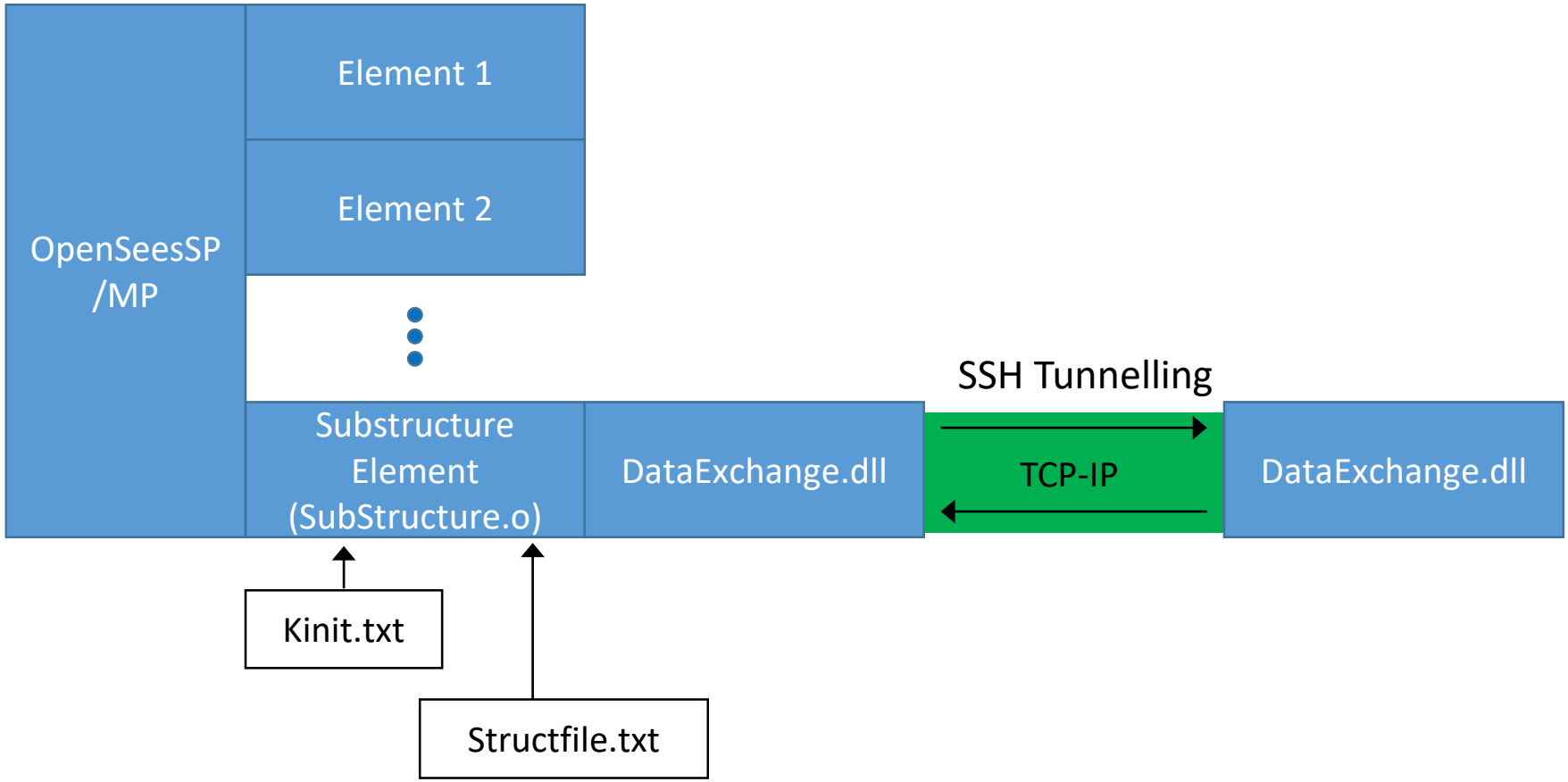
Integration Modules

- **Integration modules in UT-SIM**
  - OpenSees



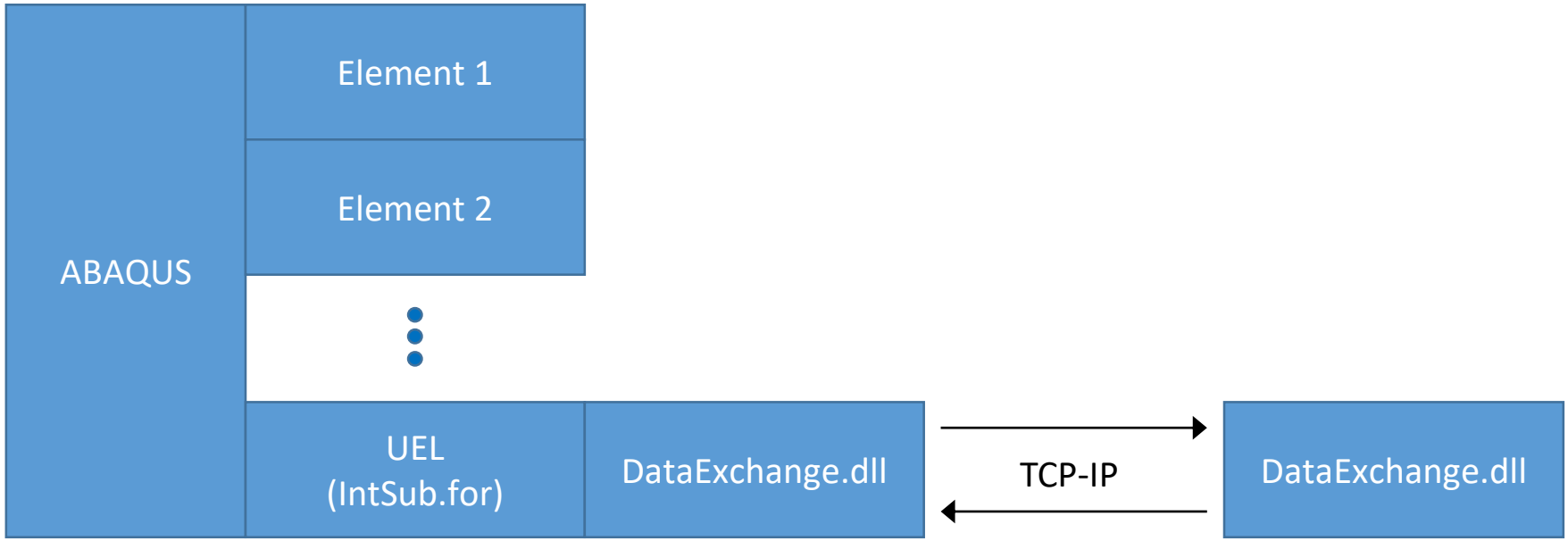
Integration Modules

- **Integration modules in UT-SIM**
  - OpenSeesSP/MP



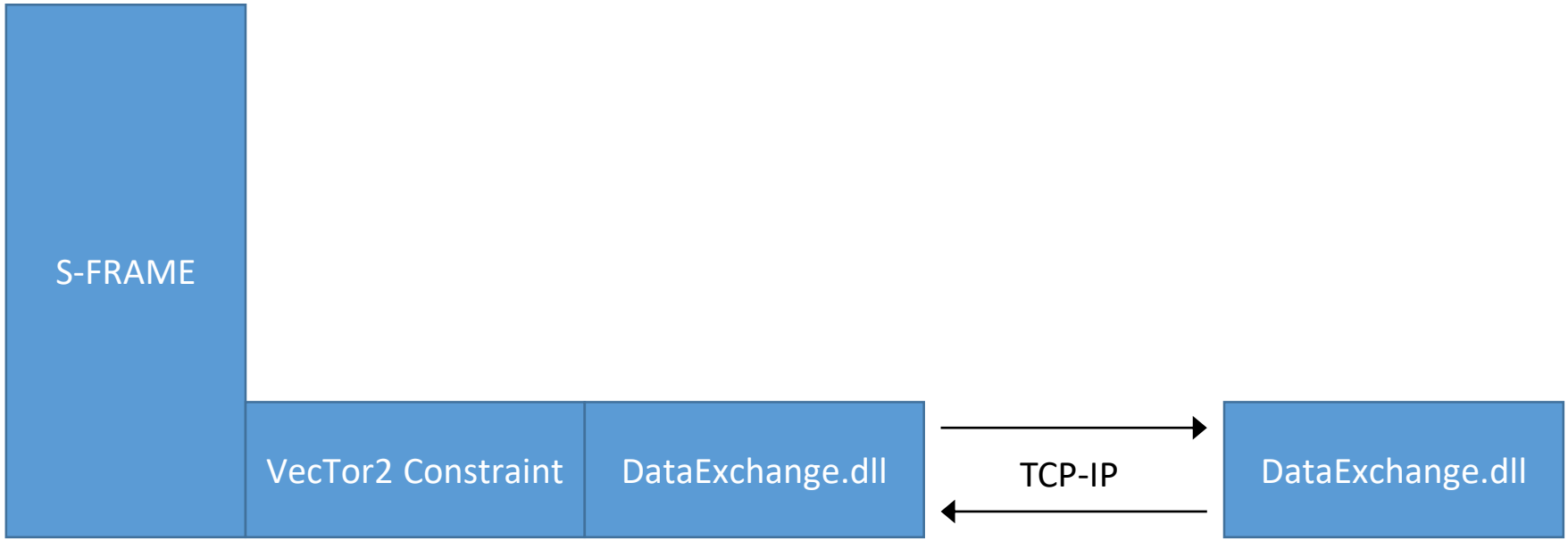
Integration Modules

- **Integration modules in UT-SIM**
  - ABAQUS



Integration Modules

- **Integration modules in UT-SIM**
  - S-Frame



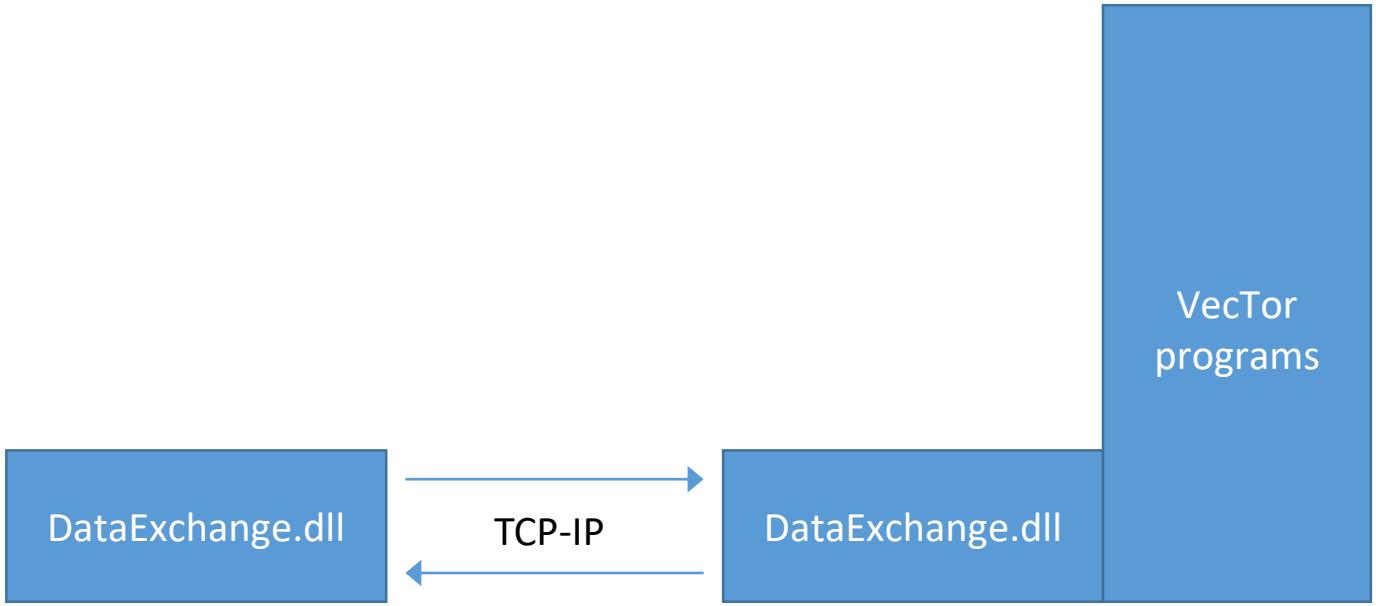
Substructure Modules

- **Main tasks**
  - Models of relatively small regions in a structural system that needs to be modelled in detail.
  - Provide the integration module with restoring forces

- **Implementation methods**
  - revise source code
  - Link element
  - Interface program
    - NICA
    - NICON

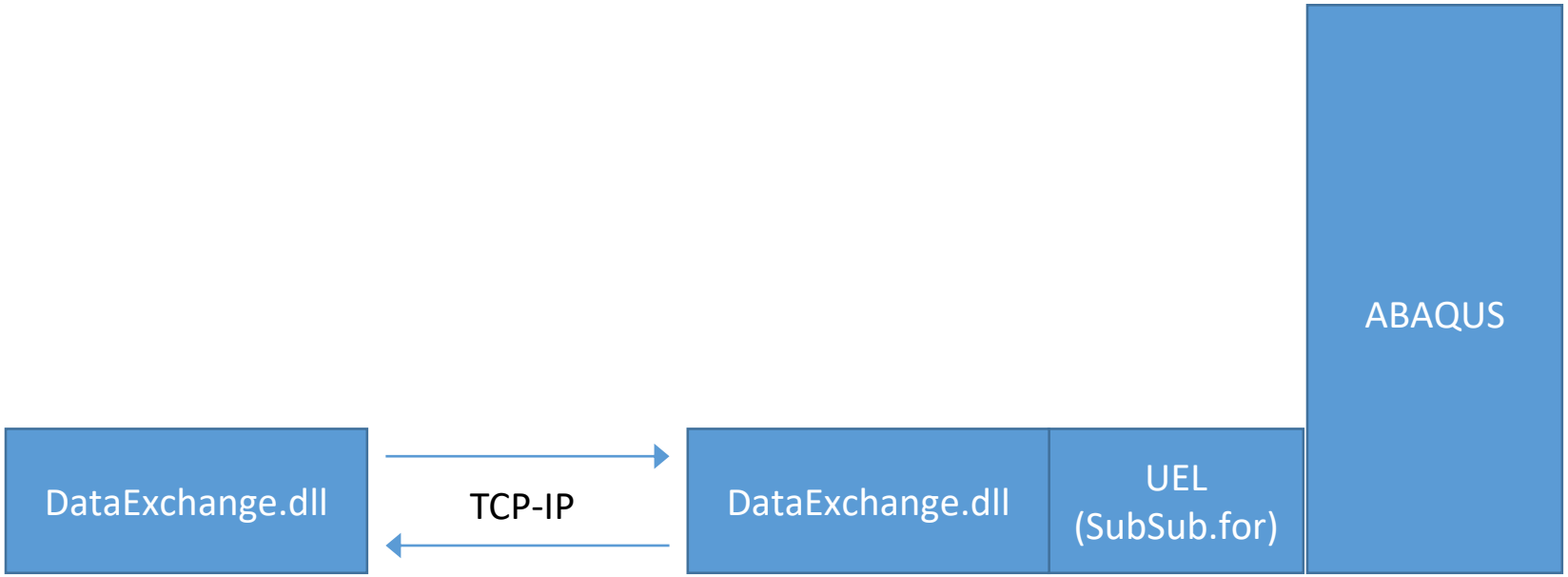
Substructure Modules

- Revise source code



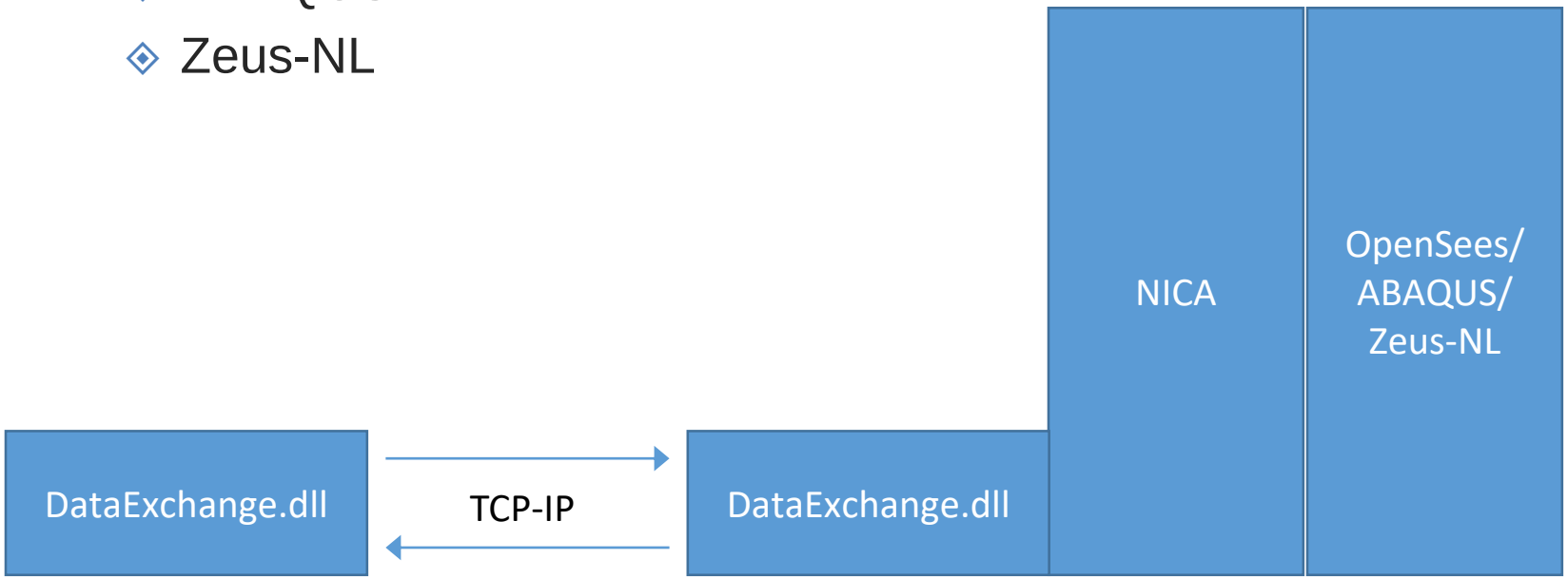
Substructure Modules

- Link element



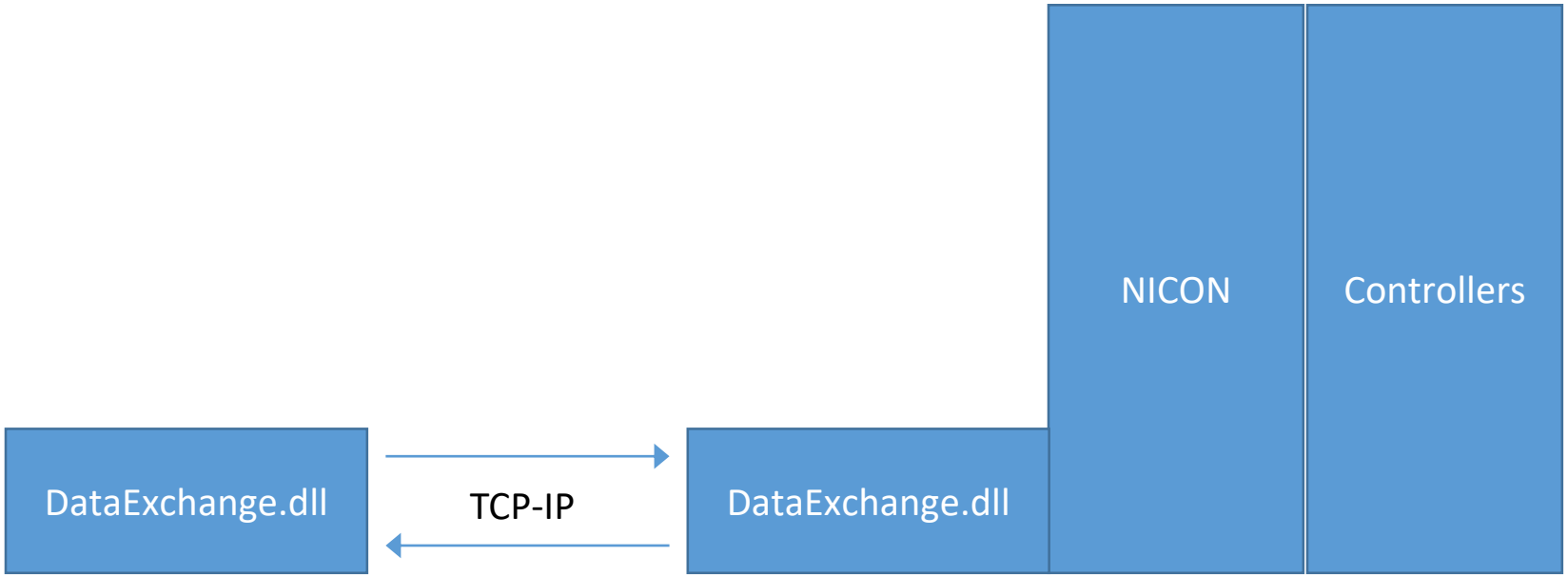
Substructure Modules

- **NICA (Network Interface for Console Applications)**
  - OpenSees
  - ABAQUS
  - Zeus-NL



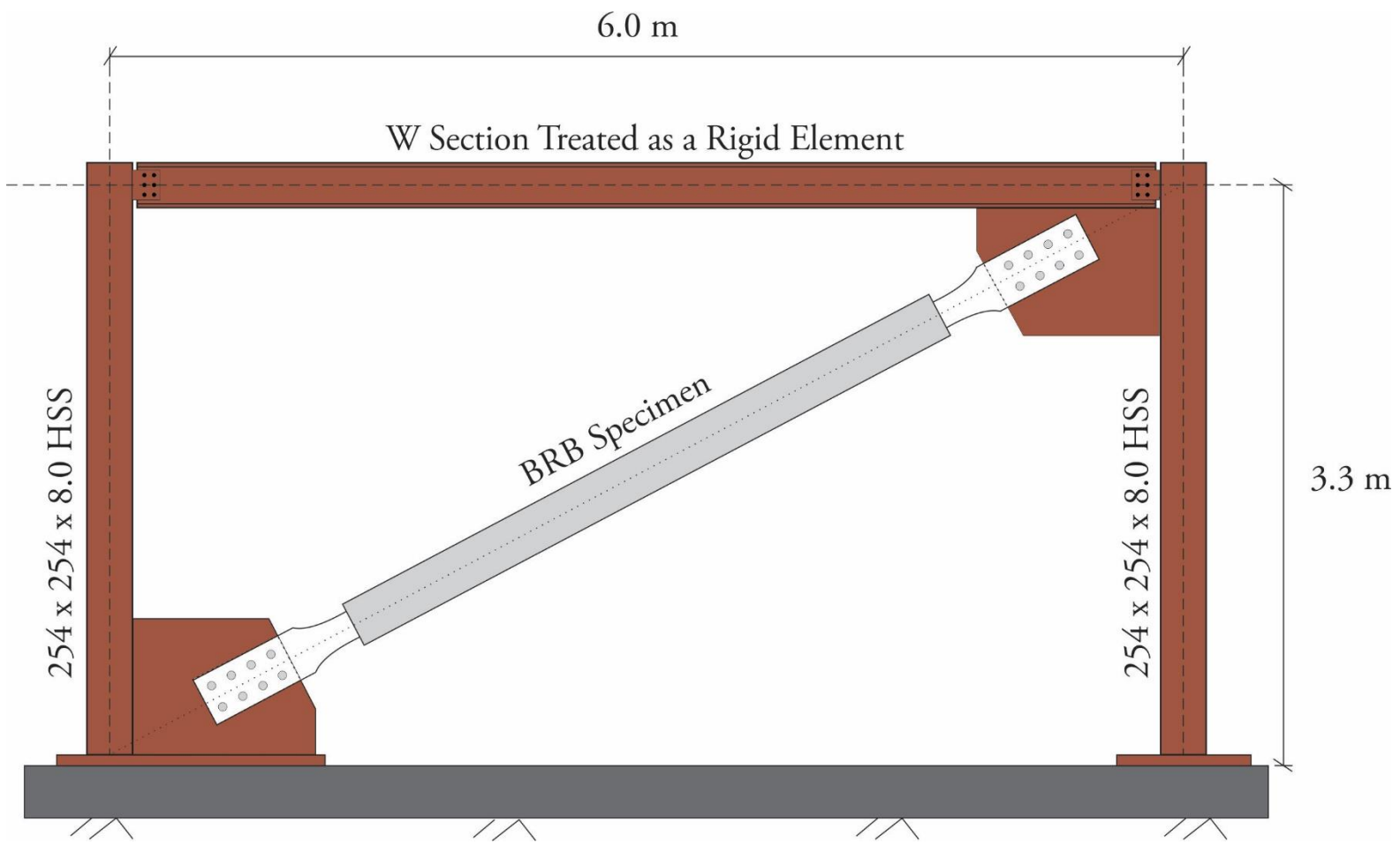
Substructure Modules

- NICON (Network Interface for Controllers)



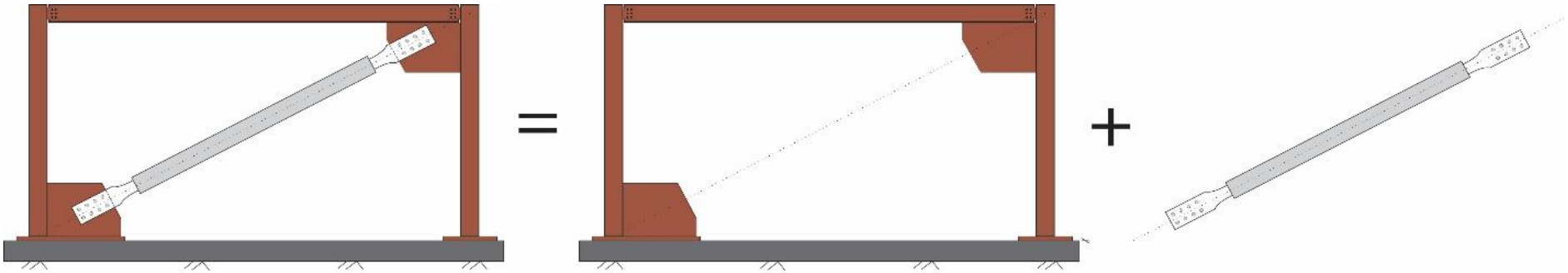
Example Structures

- **Example Structure 1**



Example Structures

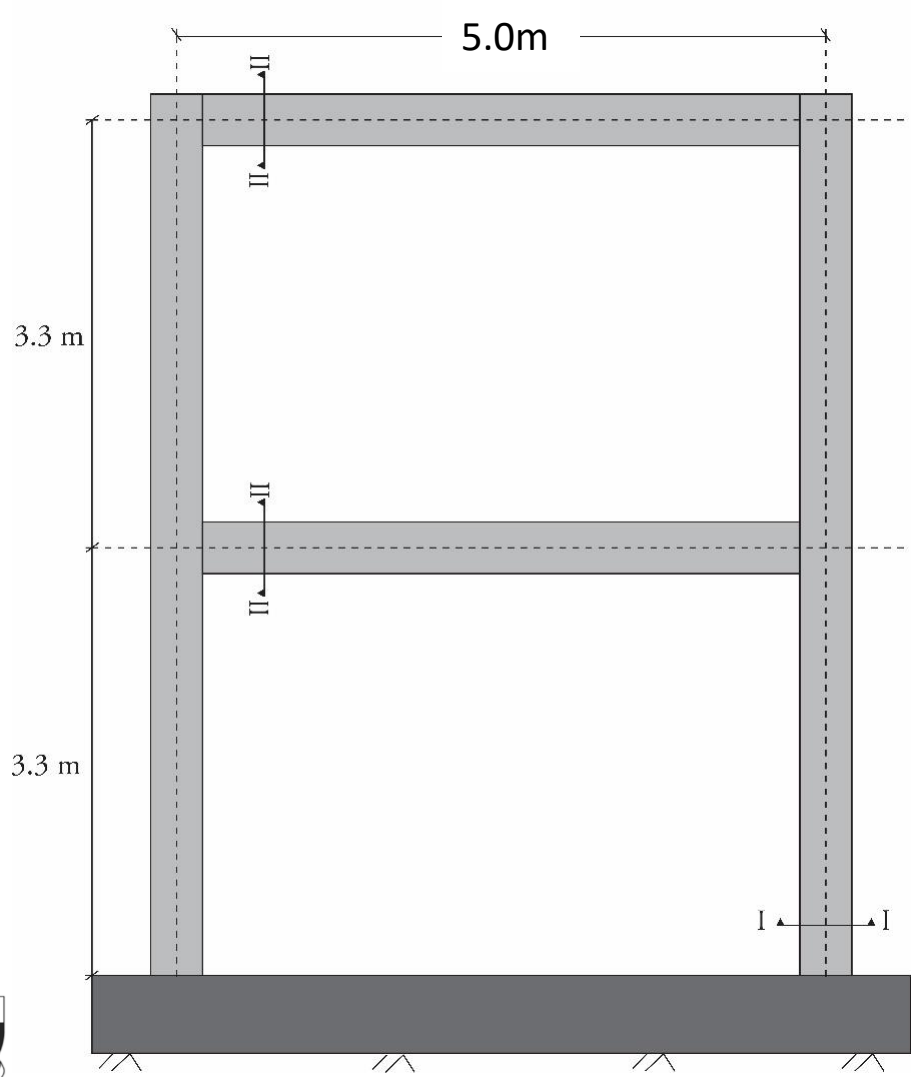
- **Example Structure 1**

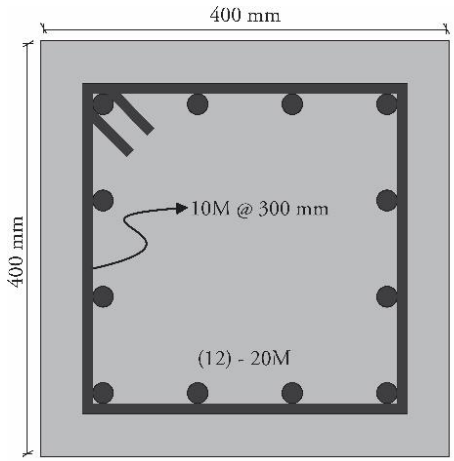


Complete | Integration module | Substructure module

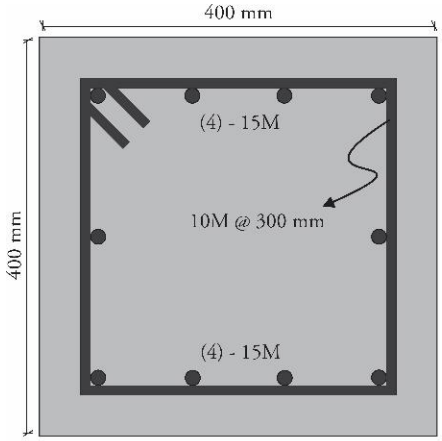
Example Structures

- **Example Structure 2**





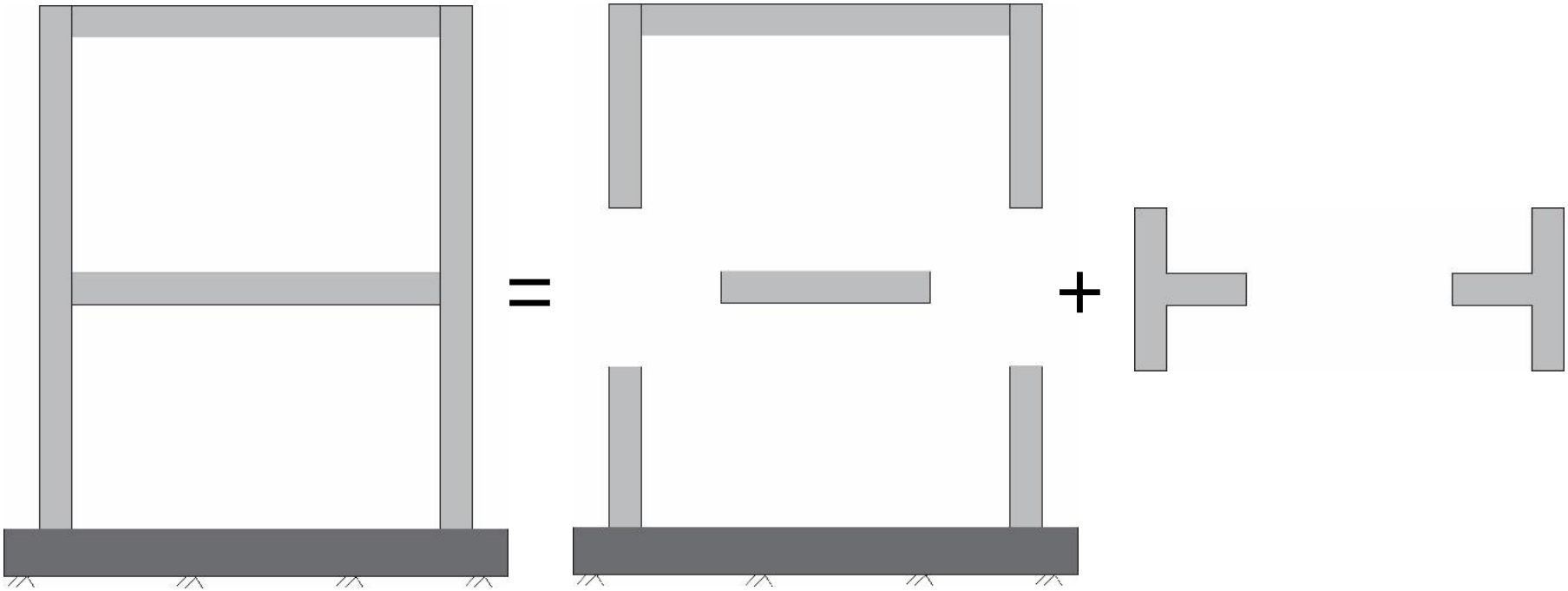
Section I-I: Column details



Section II-II: Column details

Example Structures

- **Example Structure 2**

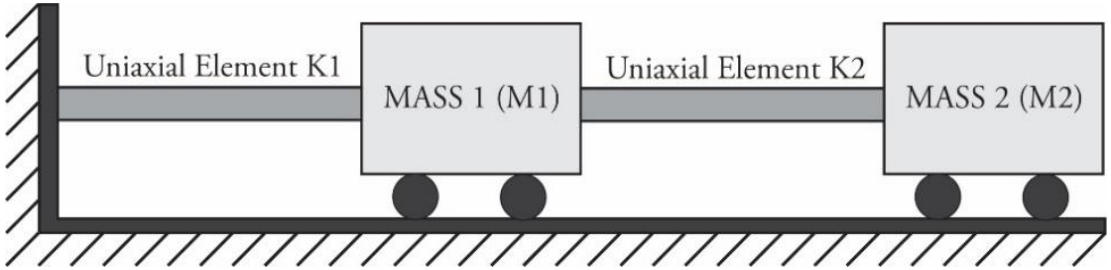


Complete | Integration module | Substructure module

Example Structures

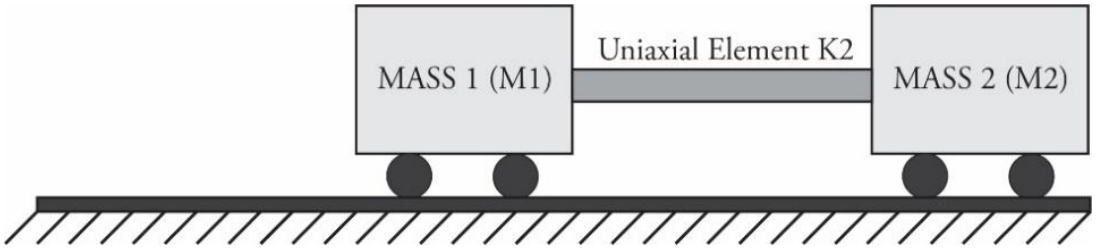
❑ Example Structure 3

Complete



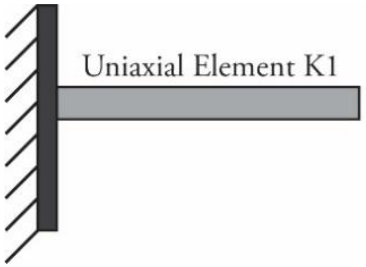
=

Integration module



+

Substructure module



Thanks for your attention!

Questions?