Session 14

Data Sharing Techniques in ASP.NET MVC

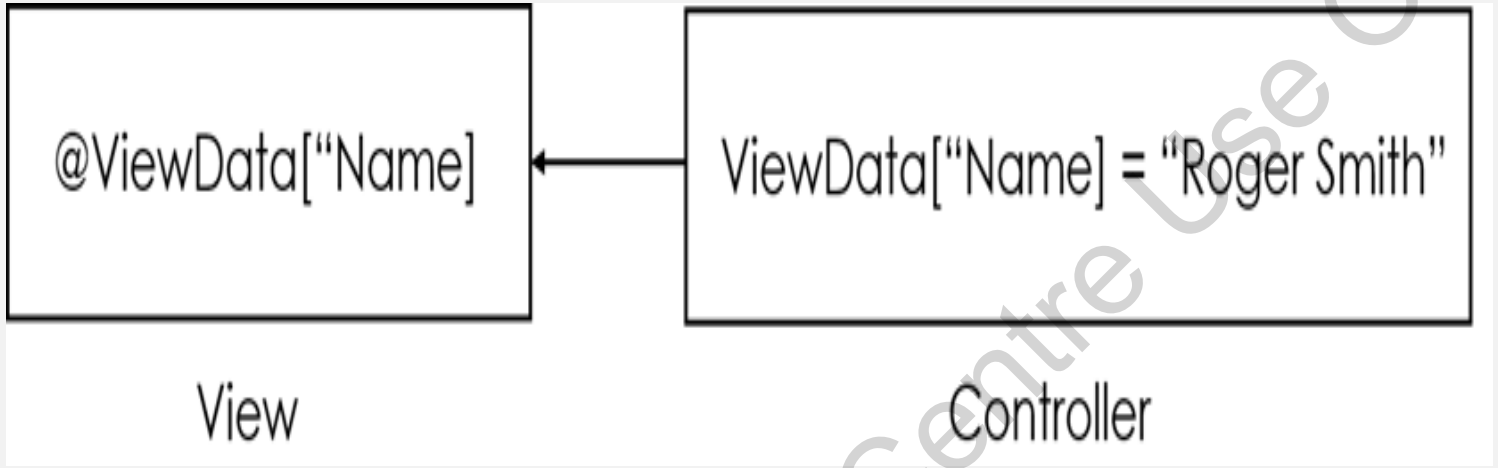
Session Overview

- Describe ViewData
- Define ViewBag
- Explain the role of TempData
- Describe strongly typed views
- List the differences between various data sharing techniques

Introduction to Data Sharing

- ViewData
- ViewBag
- TempData
- Strongly typed view

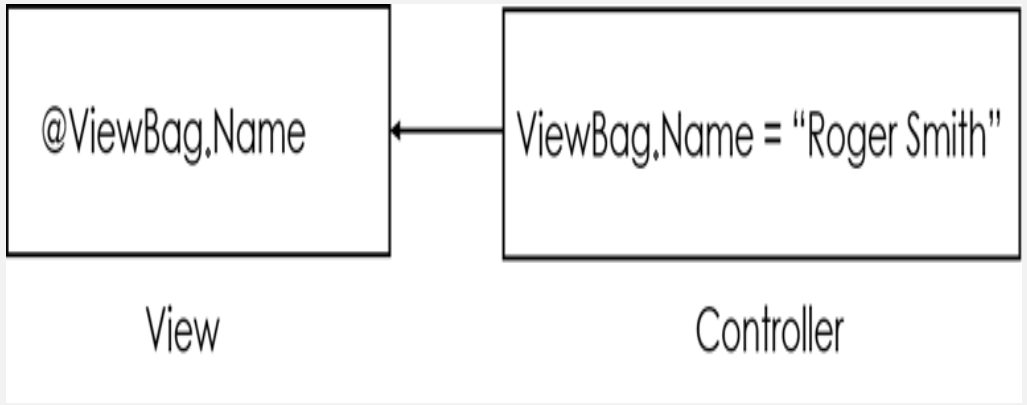
ViewData



ViewData

`ViewData` is called as a KeyValue pair object or a Dictionary object. Through the use of `ViewData`, developers can transfer data from a controller to a view. By default, the type of `ViewData` is an object of `ViewDataDictionary` class. In this Dictionary object, Strings are used as keys to store and access value or data from `ViewData`.

ViewBag (1-2)



ViewBag

ViewBag is used to transfer data from the controller to the view.

ViewBag is a dynamic data type property of the base class of all the controllers, which is the ControllerBase class.

Data stored into ViewBag is stateless

ViewBag (2-2)

- `ViewBag` can be used to transfer data which is not in the model from the controller to view.

- `ViewBag` is a dynamic data type, which internally uses `ViewData` to store values.

Use `ViewBag` to store multiple properties and values.

TempData (1-2)

- TempData is used only for current and subsequent requests as it is a very short-lived instance.

- Redirecting is the only case when users can rely on TempData.

- When redirecting, current request is killed, and a new request is created on the server to serve the redirected view.

Sharing data between the controller actions are done through the ASP.NET MVC **TempData** dictionary.

TempData value lasts until it is read or until the session times out of the current user's session.

Strongly Typed View (1-3)

Strongly typed view is the view that bonds any model with a view

Strongly Typed View (2-3)

Automatic Scaffolding

Depending on the selected template and model, it creates a view with skeleton.

IntelliSense Support

Visual Studio is used to show IntelliSense with the help of the model.

Compile Time Type Checking

Problems with data type are detected by the compiler and compiler errors are thrown rather than runtime errors.

Strongly Typed View (3-3)

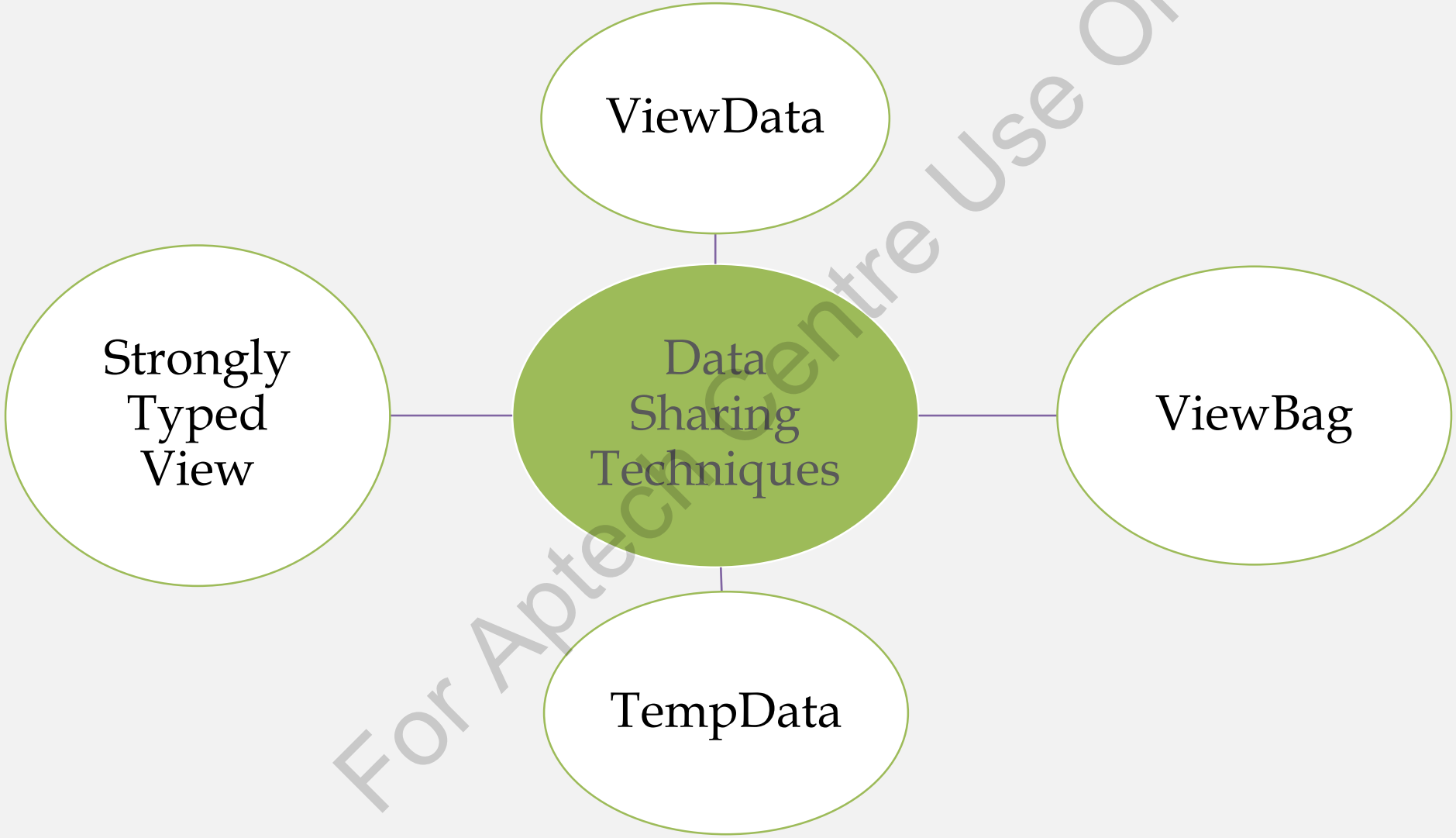
It retrieves values from ViewData.Model, rather than setting them in properties.

It supports IntelliSense and type safety.

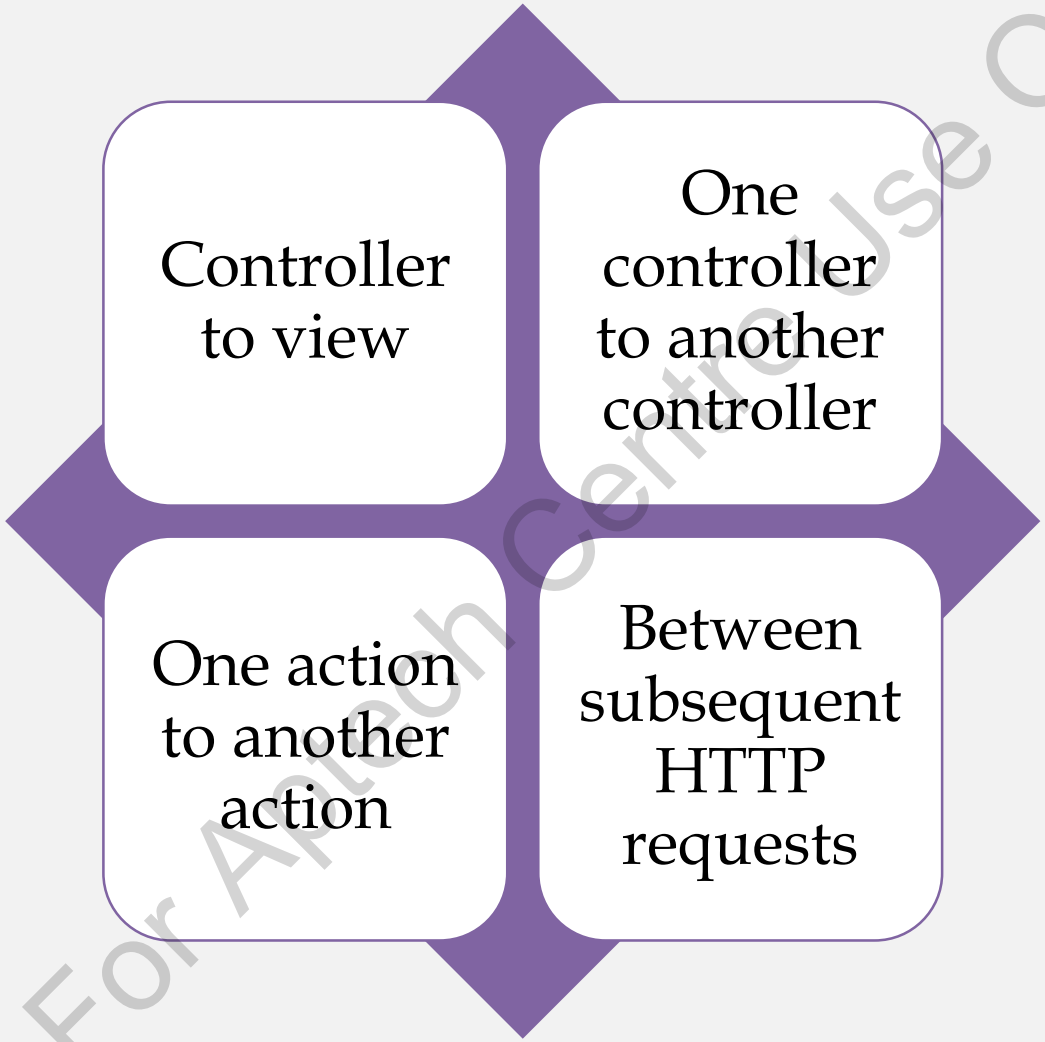
It does not include any unnecessary casting between types in ViewData.

It implements compile-time checks.

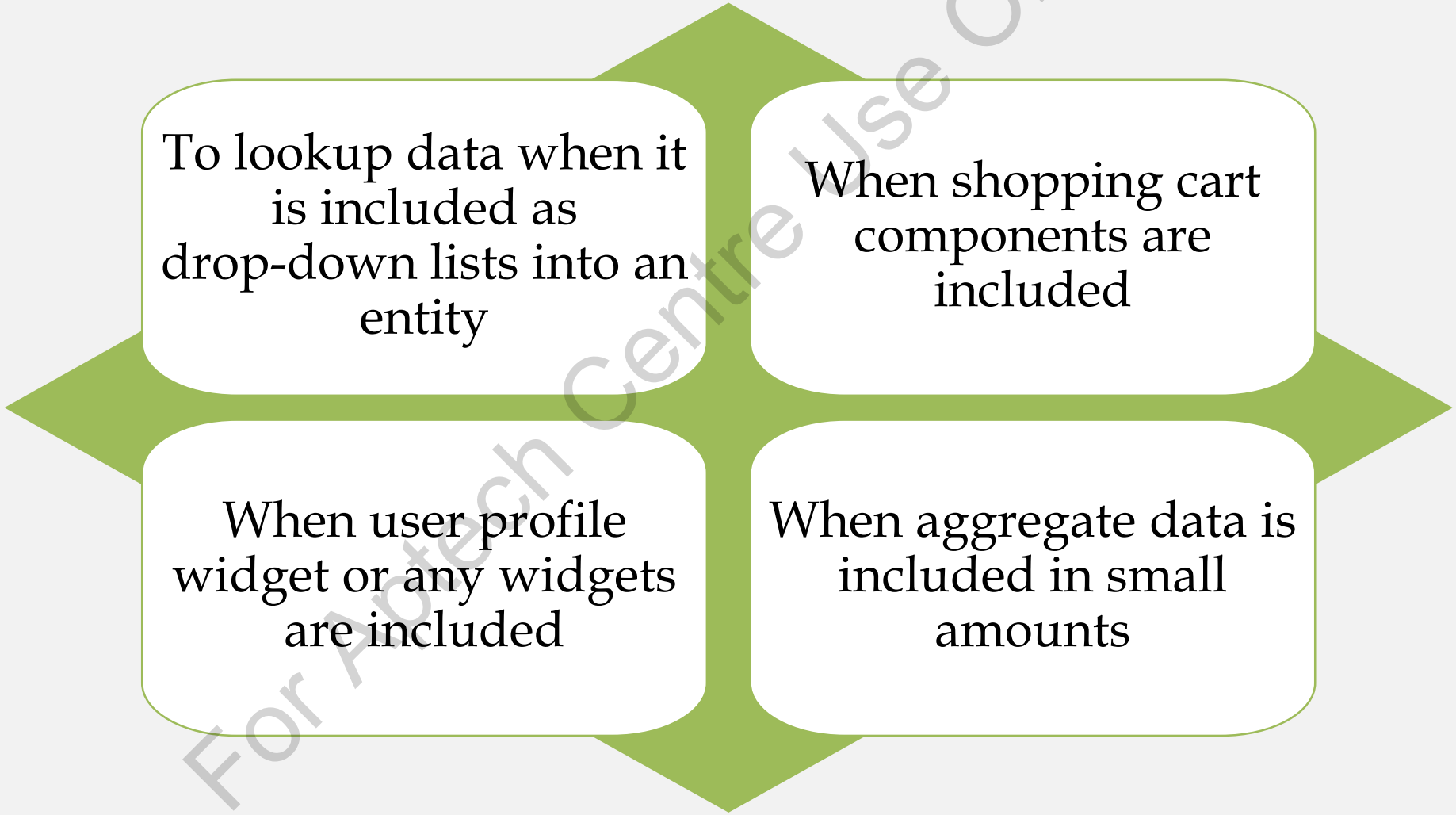
Data Sharing Techniques (1-2)



Data Sharing Techniques (2-2)



Comparing Data Sharing Techniques (1-2)



Comparing Data Sharing Techniques (2-2)

ViewData	ViewBag	TempData
It transfers data from the controller to the view. It can be accessed using strings as keys.	A dynamic wrapper around `ViewData` is `ViewBag`.	If data is required for the next request also, then use `TempData`. Note that the data will be gone after the next request.
It obtains a null value in case of redirection. However, it also requires typecasting for complex data types.	It is found only in ASP.NET MVC 3 onwards. It can handle complex data types without typecasting. It obtains a null value in case of a redirection.	Data is passed from the current request to the subsequent request (redirection) using `TempData`. Due to this reason, `TempData` value will not be null.

Comparison Between ViewData, ViewBag, and TempData

Summary

- Methods used to transfer results from a controller to a view are as follows:
  - `ViewData`
  - `ViewBag`
  - `TempData`
  - `Strongly typed view`
- `ViewData` is inherited from `ViewDataDictionary` class and is a dictionary of objects.
- To transfer temporary data from the controller to the view, a `ViewBag` is used.
- `TempData` is used to transfer data between the current and next HTTP requests.
- Strongly typed view binds any model with a view. Various advantages are displayed by strongly typed view over standard view.