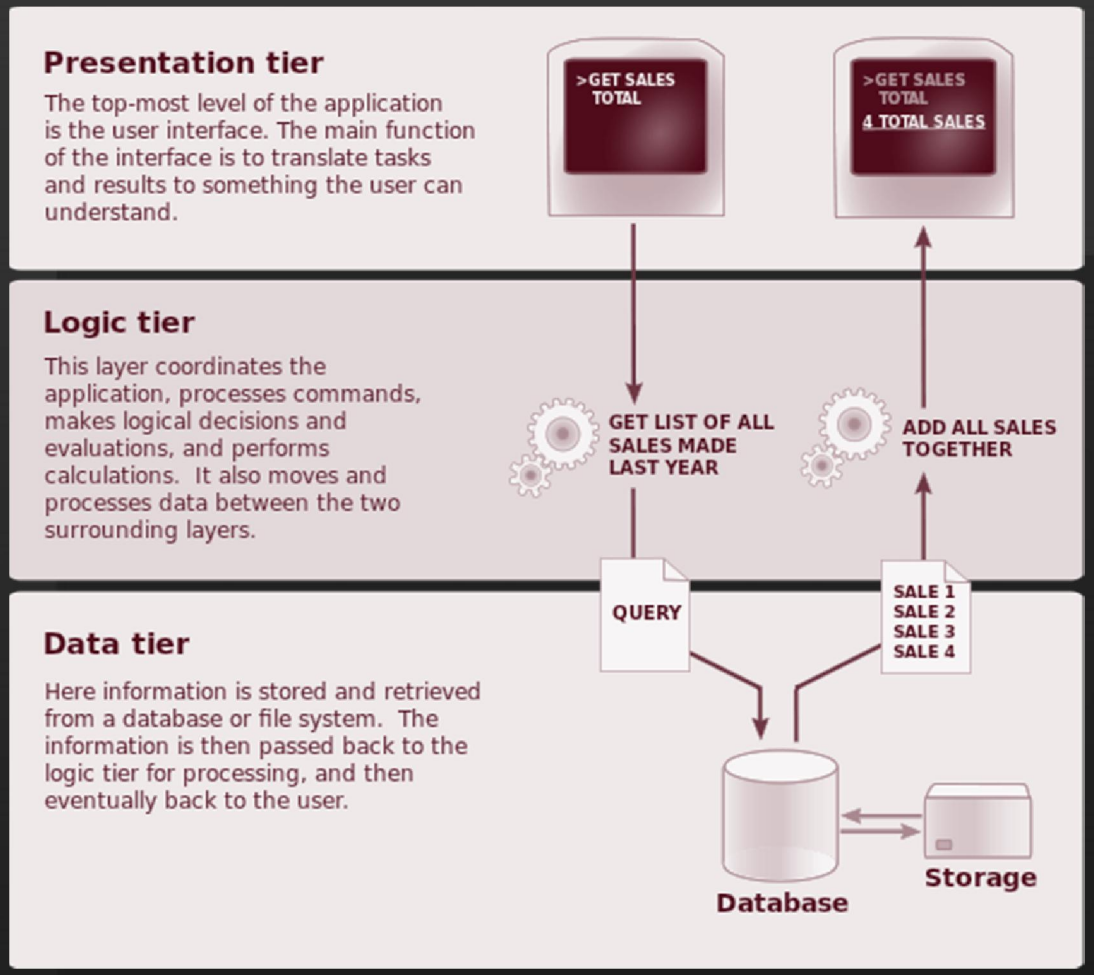
# What is ASP.NET MVC?

- A new Web Application Project type
- Simply an option
  - **Not** a replacement for WebForms
  - Builds on top ASP.NET

# What is MVC?

- A design pattern
- Acronym for Model ● View ● Controller
- Separation of concerns




*Common 3-Tier Architecture Model*

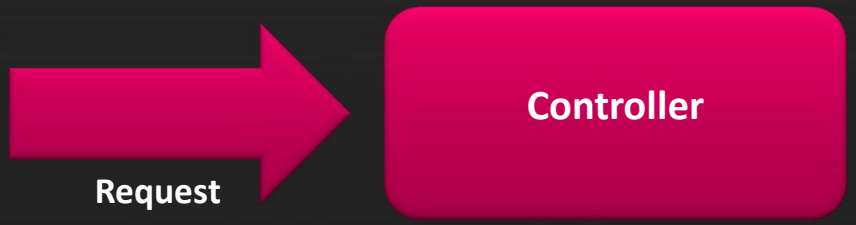
Separates representation of information from user interaction.

Promotes:

- **Code Reusability**
- Separation of Concerns

# Software Architecture Pattern

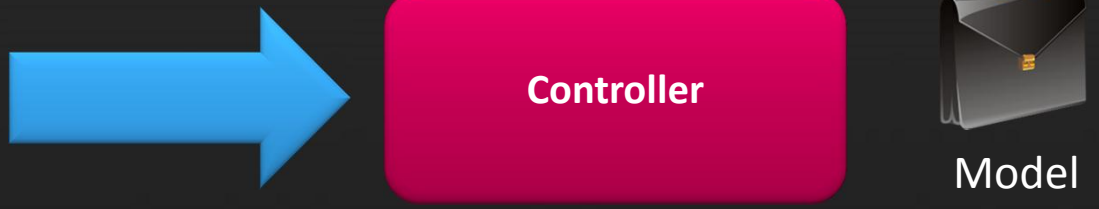
# What is MVC?




Step 1

Incoming request directed to **Controller**

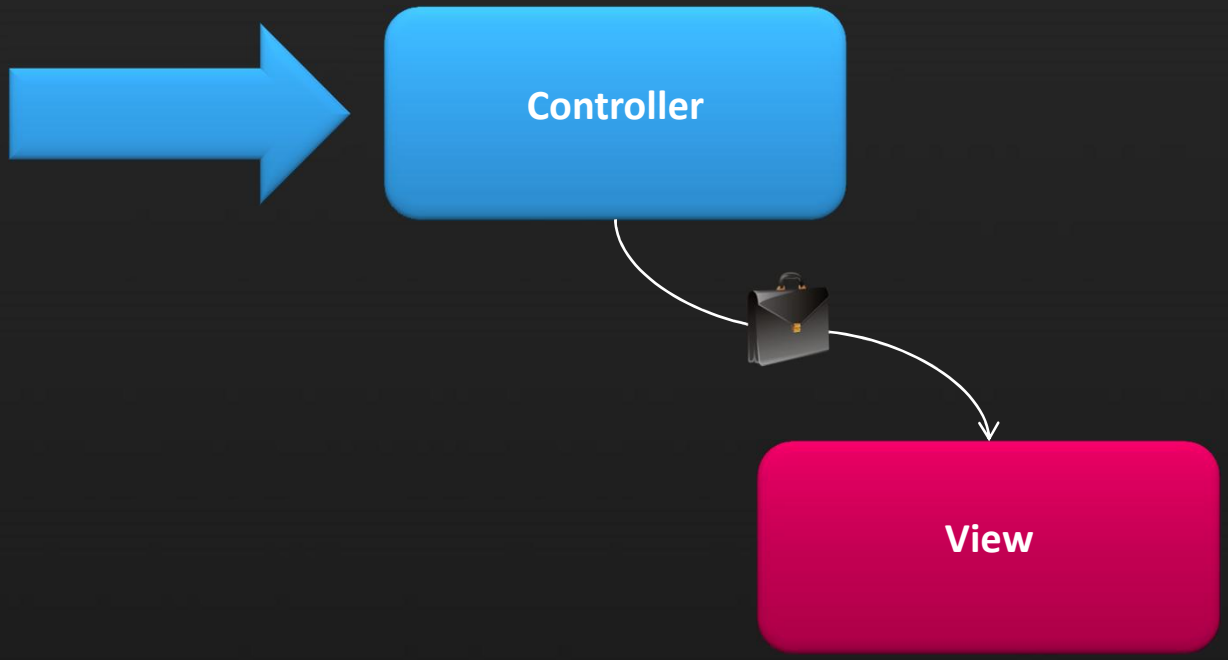
# What is MVC?




Step 2

**Controller** processes request and forms a data **Model**

# What is MVC?




Step 3

**Model** is passed to **View**

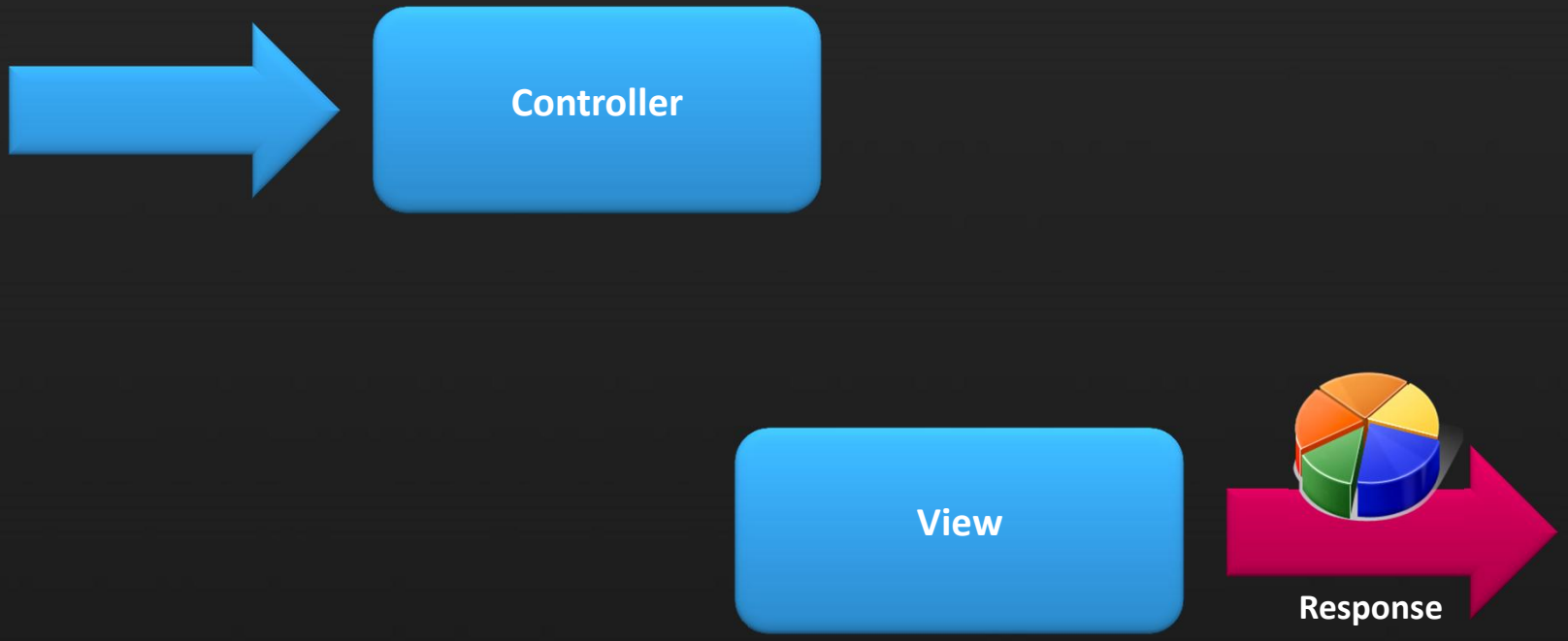
# What is MVC?



Step 4

**View** transforms **Model** into appropriate output format

# What is MVC?




Step 5

Response is rendered

Controller – Mediates input and commands for the model or view

Model – Application data, business rules, logic, and functions.

View – Output and representation of data

# Execution Process

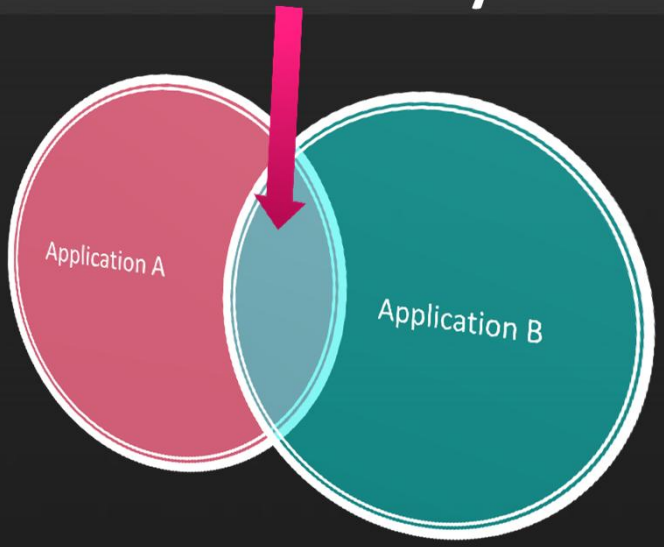
# Framework Goals

- Frictionless Testability
- Tight control over markup
- User/SEO friendly URLs
- Leverage the benefits of ASP.NET
- Conventions and Guidance

# Separation Of Concerns

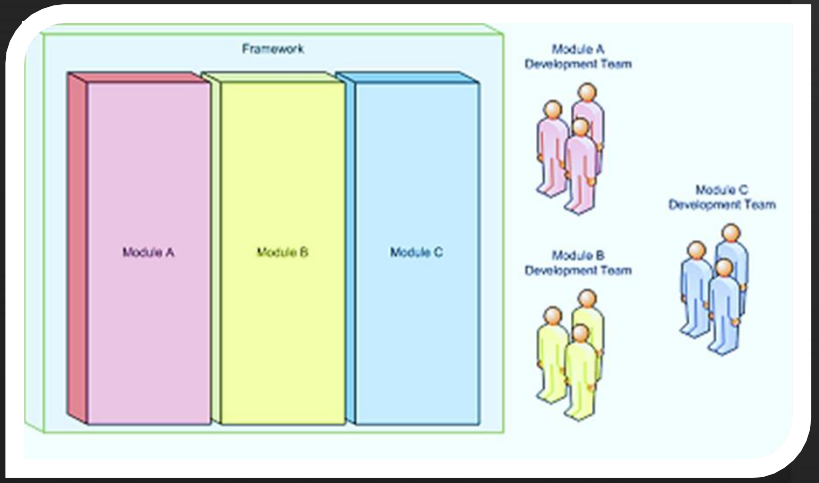
- Each component has one responsibility
  - SRP – Single Responsibility Principle
  - DRY – Don’t Repeat Yourelf
- More easily testable
- Helps with concurrent development

## Code Reusability




- Shortens development
- Code Libraries
- Design Patterns
- Frameworks

## Separation of Concerns




- Improves code clarity and organization
- Helps troubleshooting by isolating issues
- Allows for multiple teams to develop simultaneously

# Extensible

- Replace any component of the system
  - Interface-based architecture
  - Very few sealed methods / classes

# WebForms are extremely useful …

- Mature, proven technology
- Scalable
- Extensible
- Familiar feel to WinForms developers

# WebForms have challenges

- Difficult to test
- Lack of control over markup
- It does things you didn’t tell it to do

# Summary

- Not a replacement for WebForms
  - All about alternatives
- Fundamental
  - Part of the System.Web namespace
  - Same team that builds WebForms

# Summary
## It's still ASP.NET

- Providers still work
  - Membership, Caching, Session, etc.
- Views leverage .aspx and .ascx
  - But they don't have to if you don't want them to
- Within System.Web namespace
- Feature Sharing

# Summary
## Extensible

- Replace Any Part with one of your own
- As simple or complex as it needs to be to suit your tasks

# Summary
## Clean URL Structure

- Fits with the nature of the web
  - MVC exposes the stateless nature of HTTP
- Friendlier to humans
- Friendlier to web crawlers
  - Search engine optimization (SEO)

# Appendix

## Choosing Between The Two

# Use MVC if…

- You want full control over markup
- You want a framework that *enforces* separation of concerns
- TDD/Unit Testing is a priority for you
- Control abstractions get in your way more than they help
- You like writing Javascript

# Use Asp.Net WebForms if…

- You like programming against the reusable control abstraction that encapsulate UI and logic
- You like using the WYSWIG designer and would rather avoid angle brackets
- You like keeping logic on the server rather than hand writing Javascript
- Unit testing with the MVP pattern is sufficient for your needs