

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»  
(СПбГУТ)**

---

**КАФЕДРА АВТОМАТИЗАЦИИ ПРЕДПРИЯТИЙ СВЯЗИ**

**ПРЕДМЕТНО-  
ОРИЕНТИРОВАННОЕ WEB-  
ПРОГРАММИРОВАНИЕ**

**2020**

## Инструментарий

Visual Studio 2019 Community

SQL Server 2017 Express

Microsoft SQL Server Management Studio 2017

Эрик Эванс

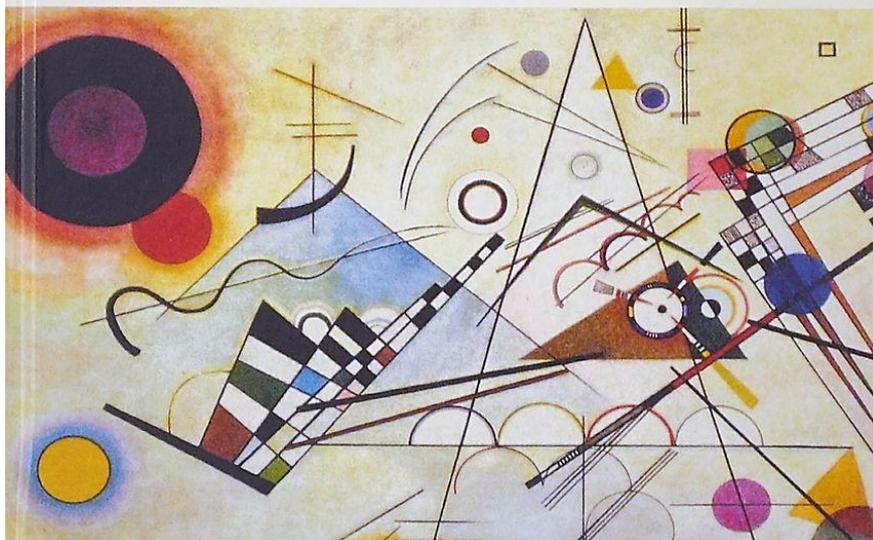
Предисловие  
Мартина Фаулера

Domain-Driven

DESIGN

# Предметно-ориентированное проектирование

СТРУКТУРИЗАЦИЯ СЛОЖНЫХ  
ПРОГРАММНЫХ СИСТЕМ



# Интернет и всемирная паутина

## Интернет и Всемирная паутина

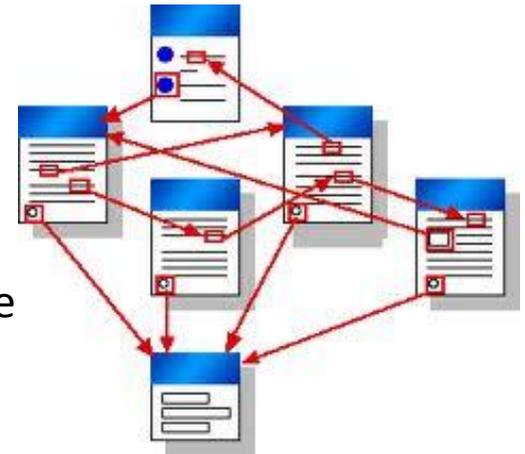
**Интернет** (англ. Internet) – всемирная система объединённых компьютерных сетей для хранения, обработки и передачи информации.

**Всемирная паутина** (англ. World Wide Web) – распределённая система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключённых к сети Интернет. Для обозначения Всемирной паутины также используют слово веб (англ. web «паутина») и аббревиатуру WWW.



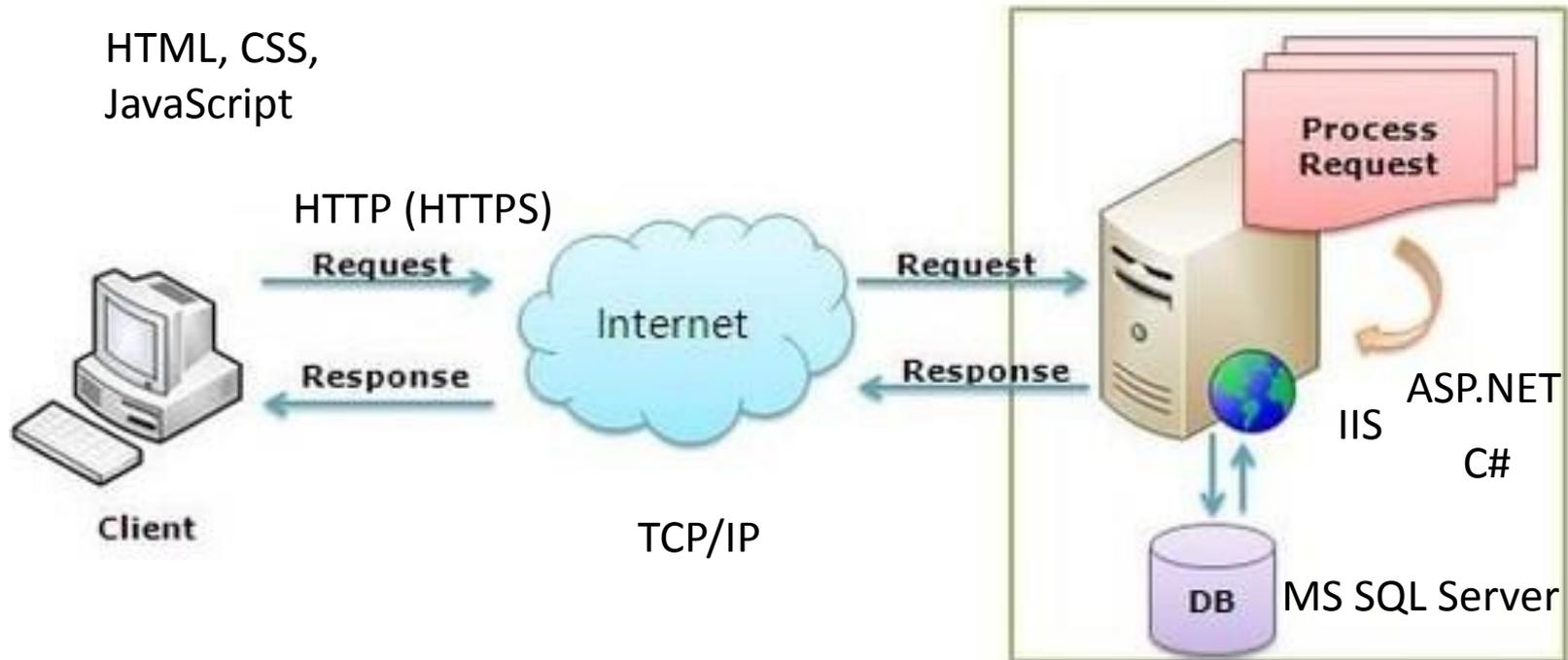
Интернет – это то, что связывает компьютерные сети в глобальном (мировом) масштабе.

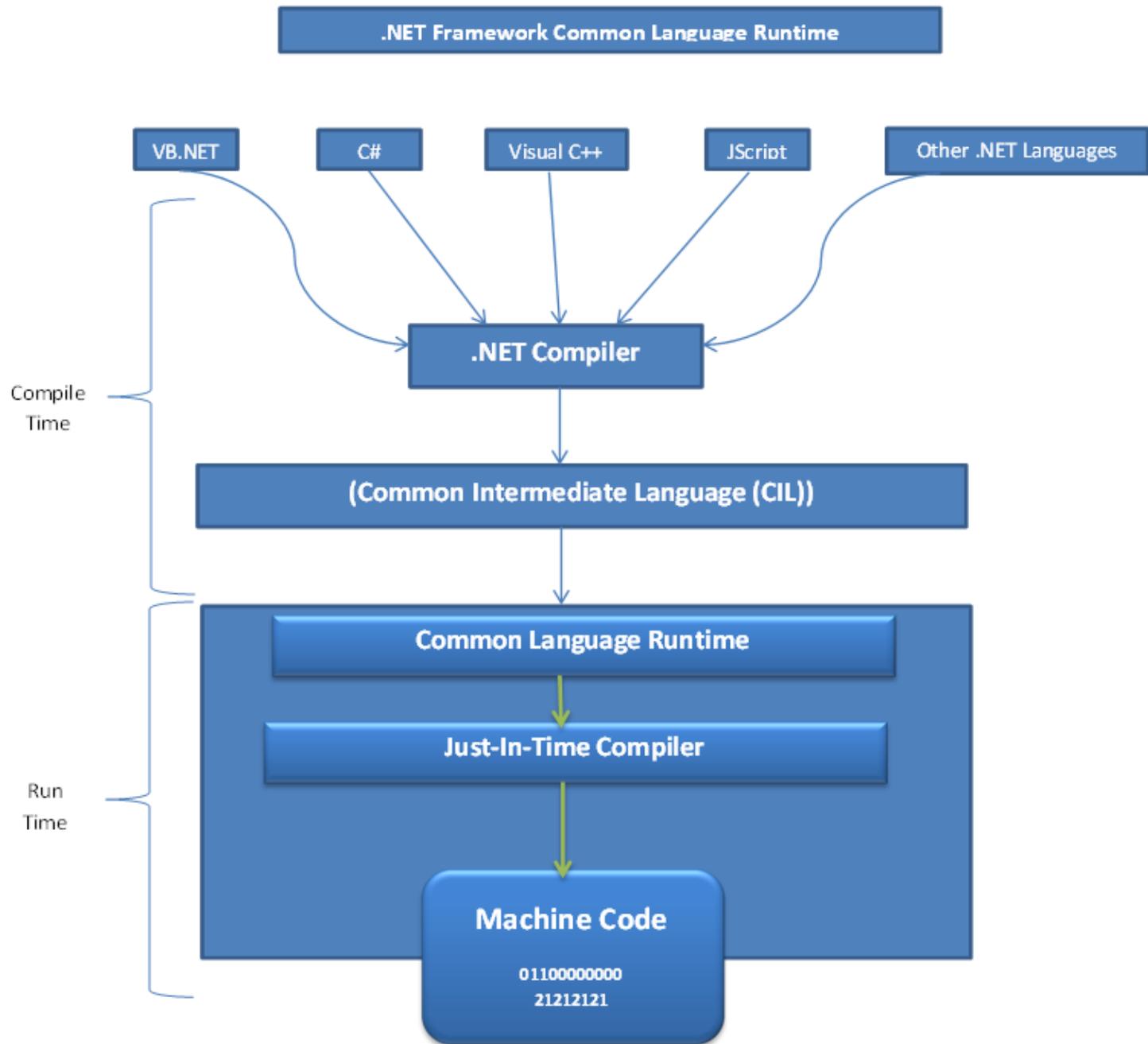
Всемирная паутина – это способ организации и обмена информацией (все гипертекстовые документы, которые хранятся на веб-серверах Интернета).

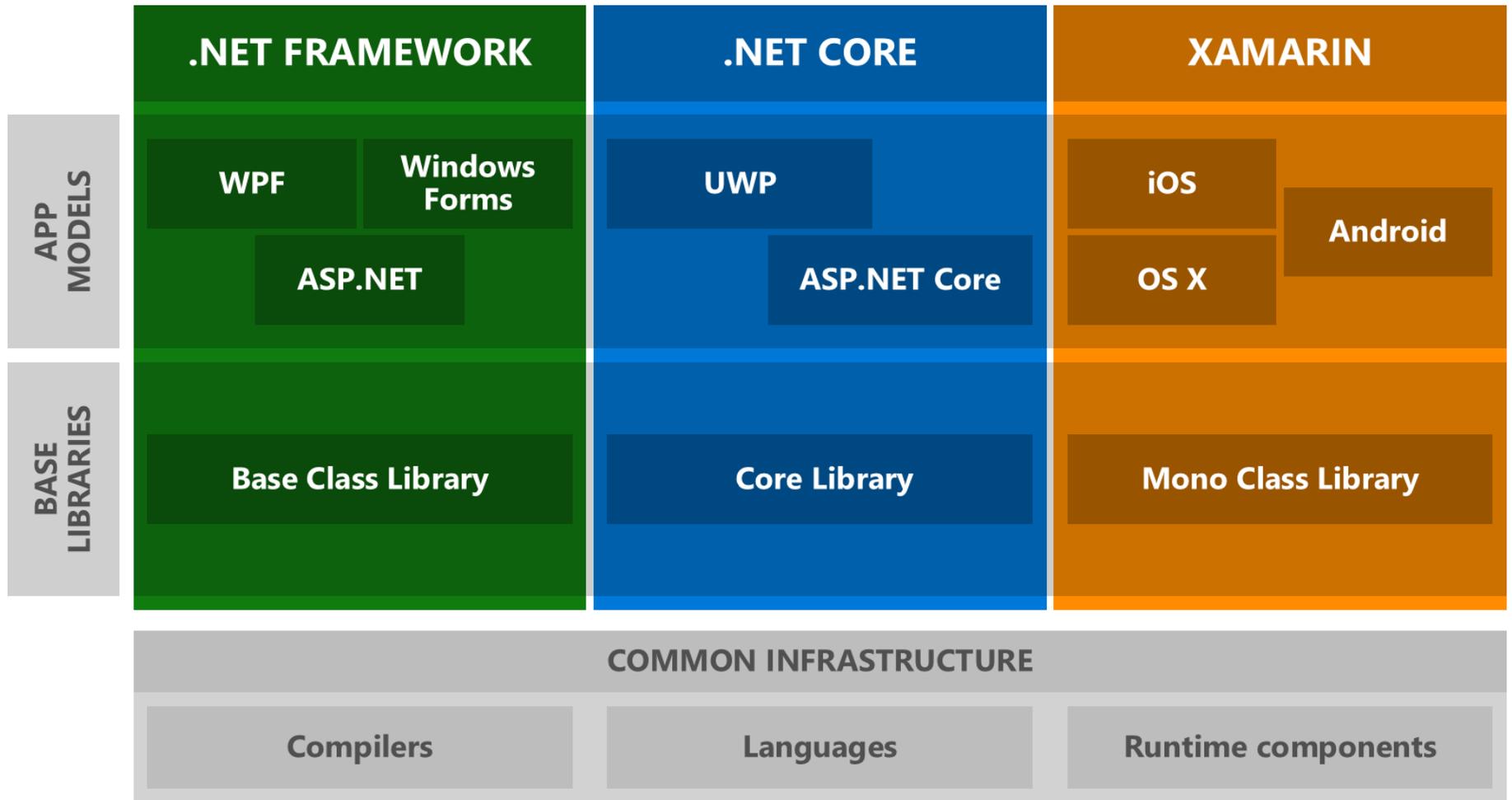


Можно сказать, что Интернет – это физическая система, а Всемирная паутина – логическая.

# Клиентские и серверные технологии







**.NET FRAMEWORK**

**.NET CORE**

**XAMARIN**

APP  
MODELS

WPF

Windows  
Forms

UWP

ASP.NET Core

iOS

Android

OS X

ASP.NET

BASE  
LIBRARIES

Base Class Library

Core Library

Mono Class Library

**COMMON INFRASTRUCTURE**

Compilers

Languages

Runtime components

```
@page
```

```
@model RazorPagesContacts.Pages.CreateModel
```

```
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```

```
<html>
```

```
<body>
```

```
  <p>
```

```
    Enter your name.
```

```
  </p>
```

```
  <div asp-validation-summary="All"></div>
```

```
  <form method="POST">
```

```
    <div>Name: <input asp-for="Customer.Name" /></div>
```

```
    <input type="submit" />
```

```
  </form>
```

```
</body>
```

```
</html>
```

```
namespace RazorPagesContacts.Pages
{
    public class CreateModel : PageModel
    {
        private readonly AppDbContext _db;

        public CreateModel(AppDbContext db)
        {
            _db = db;
        }

        [BindProperty]
        public Customer Customer { get; set; }

        public async Task<IActionResult> OnPostAsync()
        {
            if (!ModelState.IsValid)
            {
                return Page();
            }

            _db.Customers.Add(Customer);
            await _db.SaveChangesAsync();
            return RedirectToPage("/Index");
        }
    }
}
```

```
@page
@model RazorPagesContacts.Pages.IndexModel
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers

<h1>Contacts</h1>
<form method="post">
  <table class="table">
    <thead>
      <tr>
        <th>ID</th>
        <th>Name</th>
      </tr>
    </thead>
    <tbody>
      @foreach (var contact in Model.Customers)
      {
        <tr>
          <td>@contact.Id</td>
          <td>@contact.Name</td>
          <td>
            <a asp-page="./Edit" asp-route-id="@contact.Id">edit</a>
            <button type="submit" asp-page-handler="delete"
              asp-route-id="@contact.Id">delete</button>
          </td>
        </tr>
      }
    </tbody>
  </table>

  <a asp-page="./Create">Create</a>
</form>
```

```
namespace RazorPagesContacts.Pages
{
    public class IndexModel : PageModel
    {
        private readonly AppDbContext _db;

        public IndexModel(AppDbContext db)
        {
            _db = db;
        }

        public IList<Customer> Customers { get; private set; }

        public async Task OnGetAsync()
        {
            Customers = await _db.Customers.AsNoTracking().ToListAsync();
        }

        public async Task<IActionResult> OnPostDeleteAsync(int id)
        {
            var contact = await _db.Customers.FindAsync(id);

            if (contact != null)
            {
                _db.Customers.Remove(contact);
                await _db.SaveChangesAsync();
            }

            return RedirectToPage();
        }
    }
}
```

```
@page "{id:int}"
@model RazorPagesContacts.Pages.EditModel
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers

@{
    ViewData["Title"] = "Edit Customer";
}

<h1>Edit Customer - @Model.Customer.Id</h1>
<form method="post">
    <div asp-validation-summary="All"></div>
    <input asp-for="Customer.Id" type="hidden" />
    <div>
        <label asp-for="Customer.Name"></label>
        <div>
            <input asp-for="Customer.Name" />
            <span asp-validation-for="Customer.Name" ></span>
        </div>
    </div>

    <div>
        <button type="submit">Save</button>
    </div>
</form>
```