**Microsoft Press eBook NET Technology Guide for Business Applications**

**Mục lục**

[1. Keytakeaways 1](#_Toc479355940)

[2. Mục đích của quyển sách này 2](#_Toc479355941)

[Ai nên sử dụng quyển sách này 2](#_Toc479355942)

[Cách sử dụng quyển sách này 2](#_Toc479355943)

[3. Tổng quan 3](#_Toc479355944)

[The .NET Framework and the future of development 5](#_Toc479355945)

[4. Emerging application patterns 8](#_Toc479355946)

[Devices 11](#_Toc479355947)

[Services 15](#_Toc479355948)

[Cloud and hybrid-cloud 16](#_Toc479355949)

[End-to-end scenarios in emerging application patterns 18](#_Toc479355950)

# Keytakeaways

**Chọn phương pháp tiếp cận mô hình và công nghệ phát triển dựa trên các quyền ưu tiên và yêu cầu cho ứng dụng của bạn.**

Một mô hình và phương pháp tiếp cận duy nhất sẽ không làm việc trên tất cả các thể loại ứng dụng. Microsoft phát triển Stack và .NET rất linh hoạt và cung cấp nhiều khả năng, nhưng điều quan trọng đó là bạn chọn phương pháp tiếp cận và công nghệ dựa trên các loại ứng dụng hoặc cả hệ thống phụ mà bạn xây dựng nên. Mỗi ứng dụng sẽ có sự ưu tiên khác nhau và sự đánh đổi phải được thực hiện trên con đường khác nhau.

**Hiện đại hóa ứng dụng nghiệp vụ đi xa hơn đơn giản chỉ là việc xây dựng các ứng dụng di động. Ứng dụng di động phải tin cậy và mở rộng ứng dụng nghiệp vụ của bạn.**

Để đi đến thành công, app di động phải được xây dựng với sự tích hợp sâu nền tảng nghiệp vụ hiện tại của bạn vào ứng dụng. App di động phải là một phần lớn của hệ sinh thái doanh nghiệp và có tính chất quan trọng mở rộng nền tảng nghiệp vụ ứng dụng, dù các nền tảng hệ thống là ứng dụng thiết lập và mới, lớn, nhiệm vụ quan trọng là xây dựng ứng dụng với tính sáng tạo, có khả năng mở rộng và linh hoạt.

**Định vị một ứng dụng hoặc hệ thống phụ trong phân đoạn khuôn khổ tổng quát sẽ giúp cho bạn lựa chọn đúng phương pháp và công nghệ.**

Nó là trọng tâm để định vị ứng dụng hoặc hệ thống phụ của bạn trong vùng phải phân chia. Phương pháp và công nghệ phù hơp cho từng loại ứng dụng có khả năng sẽ khác nhau:

* Mô hình ứng dụng mới nổi
  + Thiết bị và dịch vụ
* Thiết lập mô hình ứng
  + Ứng dụng nghiệp vụ vừa và nhỏ.
  + Lớn, ứng dụng nghiệp vụ trọng yếu.

# Mục đích của quyển sách này

Quyển sách này sẽ giúp bạn lựa chọn đúng công nghệ phát triển của Microsoft và các phương pháp tiếp cận cho việc phát triển ứng dụng của .NET.

Quyển sách này không đề cập đến các phương thức Quản lý vòng đời ứng dụng - Application Lifecycle Management (ALM). Để có thêm hướng dẫn về chủ đề này, bạn có thể truy cập vào trang web của Visual Studio ALM tại [www.microsoft.com/visualstudio/alm](http://www.microsoft.com/visualstudio/alm).

## **Ai nên sử dụng quyển sách này**

Quyển sách này sẽ hữu ích cho các nhà hoạch định chính sách, các kiến trúc sư phần mềm, các nhà lãnh đạo phát triển (development leads) và các nhà phát triển (developers) tham gia vào việc lựa chọn các công nghệ để sử dụng cho các ứng dụng và dự án của họ dựa trên nền tảng phát triển của Microsoft.

Quyển sách này không đề cập đến các giải pháp phát triển dựa trên các sản phẩm của Microsoft về business-market products, chẳng hạn như các giải pháp theo chiều dọc dựa trên Dynamics CRM hoặc Dynamics ERP.

## **Cách sử dụng quyển sách này**

Quyển sách này bao gồm một loạt các lựa chọn phát triển phần mềm tập trung vào các business applications. Nó được viết như một tài liệu tham khảo để bạn có thể tiếp cận đến một lĩnh vực mà bạn quan tâm, chẳng hạn như Phần 4, “Emerging application patterns” hoặc “Large, mission-critical business applications” trong Phần 5 “Established application patterns.”

Chúng tôi khuyên bạn nên đọc Mục 3 “Tổng quan (Overview)” để biết ngữ cảnh trước khi bạn đi sâu vào từng phần riêng lẻ.

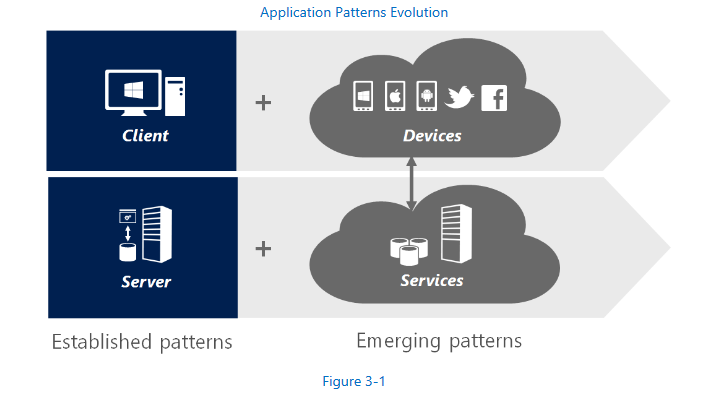
# Tổng quan

Ngày nay, Công nghệ đang được dùng theo hướng đa thiết bị được hỗ trợ bởi các dịch vụ đám mây. Các partern mẫu tăng liên tục về khả năng phần cứng như chạm, các cảm biến, và tính di động, kết hợp với sức mạnh của kết nối với web và các dịch vụ back-end như data storage, media streaming, và social connectivity.

Mối liên kết giữa các thiết bị và các dịch vụ giúp mở rộng việc kinh doanh và các kịch bản tiêu dùng. Trong không gian tiêu dùng, ban đầu điện thoại di động tạo ra một làn sóng của các thiết bị tập trung vào tiêu dùng, mà tiếp tục phát triển về khả năng phần cứng và các công nghệ nâng cao. Trong doanh nghiệp, hiện tượng song sinh giữa tiêu dùng và IT và bring-your-own-device (BYOD) đã tạo ra một động lực cho các trải nghiệm người tiêu dùng đang thúc đẩy tương lai của điện toán trong kinh doanh và các ứng dụng line-of-business (LOB).

Thế hệ tiếp theo của thiết bị - và các ứng dụng phụ thuộc dịch vụ không phải lad sự nổi lên trong cô độc. Các ứng dụng này phải làm việc trong một môt trường tích hợp cực kỳ tốt với các ứng dụng hiện có, tạo các giá trị cho các đối tượng mới và các chế độ tương tác mới. Như hình 3 – 1, tạo ra 2 pattern khác nhau mà mọi developer ứng dụng bây giờ phải đối mặt:

* Established application patterns: Đây là những ứng dụng được phát triển dùng các pattern công nghệ như client/server hoặc các ứng dụng web được tối ưu hóa cho các trình duyệt máy tính để bàn. Chúng hoạt động như các ứng dụng nền tảng và tập trung nhiều vào các business processes đã tồn tại.
* Emerging application patterns: các pattern chẳng hạn đa thiết bị và đám mây đang nổi lên như các công nghệ giúp tạo ra các ứng dụng mới. Chúng bổ sung cho các pattern đã có bằng cách mở rộng các ứng dụng để tập trung vào người dùng cuối.

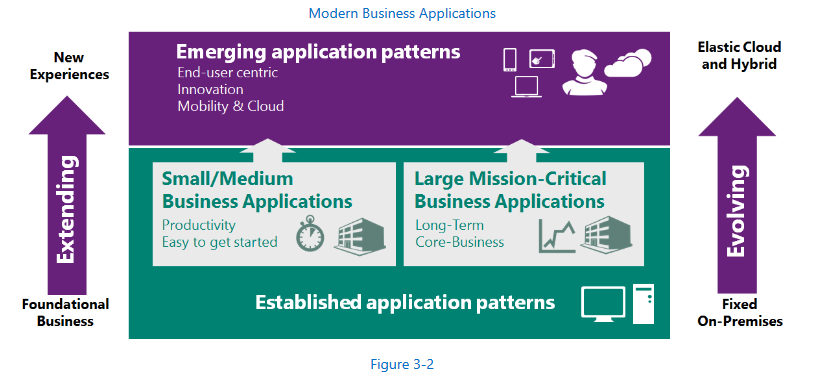


Sự xuất hiện của các pattern này để đáp ứng người dùng cuối là một cơ hội tốt cho các developer đổi mới và tạo sự khác biệt với đối thủ cạnh tranh. Bán lẻ, truyền thông, tài chính, hậu cần, các dịch vụ khác hàng – mỗi công ty là một công ty phần mềm trong thế giới kinh tế ngày nay. Mỗi công ty khả năng đáp ứng những gì khách hàng cần và cạnh tranh hiệu quả là chỉ những điều tốt nhất cho họ để cung cấp các giải pháp đổi mới phần mềm.

Tuy nhiên, mở rộng các ứng dụng hiện có để nắm lấy những nhu cầu mới là một quá trình chuyển đổi đầy thách thức. Hiện nay phát triển các công nghệ là cốt yếu theo pattern và rất khó để tích hợp các pattern mới cần thiết cho hiện đại hóa phần mềm. Các công cụ hiện tại không cung cấp một con đường rõ ràng từ thế giới client/server đến nhưng thiết bị mới nổi/ điện toán đám mây của thế giới.

Nền tảng Microsoft cho phép các developer giải quyết các thách thức này. Nó được xây dựng trên những ứng dụng hiện có, mở rộng chúng từ những pattern ứng dụng mới nổi. Nó bao trùm nhiều công nghệ phát triển, vì vậy các deverlop có thể chọn các tùy chọn phù hợp nhất với các kỹ năng và các công nghệ được dùng bởi chương trình đang tồn tại. Để phát triển các dịch vụ, Microsoft Window Azure hỗ trợ vô số công nghệ mà mất kỳ developer có thể dùng, như Java, NodeJS, PHP, Python, Ruby, hỗ trợ .NET. Phát triển client cho nền tảng Microsoft cũng hỗ trợ một loạt các công nghệ, như .NET, HTML5/JavaScript, và C++.

Tài liệu này tập trung dựa trên phát triển .NET và về các ứng dụng kinh doanh cụ thể. Nó bao gồm cách sử dụng .NET để phát triển cho các pattern mới thành lập để định hình các ứng dụng hiện có và cũng làm thể nào để nắm lấy các mô hình mới nổi cho phép tạo ra các ứng dụng kinh doanh hiện đại của tương lai; Xem hình 3-2

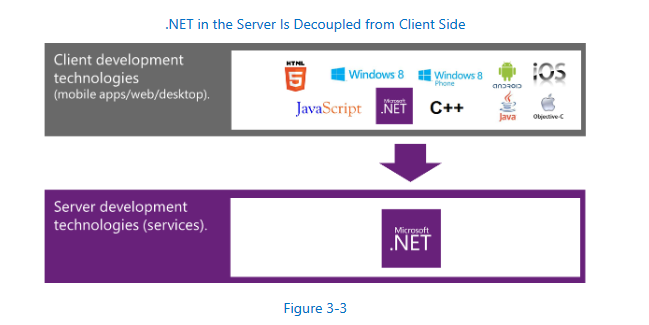


## **The .NET Framework and the future of development**

Microsoft .NET Framework được xây dựng để cho phép các nhà phát triển tạo ra các ứng dụng dựa trên nền tảng Microsoft, và bằng tất cả các tài khoản, nó đã là một thành công lớn trên thị trường. Ngày nay, hàng triệu nhà phát triển trên khắp các công ty thuộc mọi quy mô và phân đoạn dựa vào .NET để tạo ra các ứng dụng. Nó cung cấp các dịch vụ cốt lõi cần thiết để xây dựng các ứng dụng của người tiêu dùng; Các ứng dụng kinh doanh nhỏ; Và các ứng dụng lớn, quan trọng, tất cả đều có chất lượng, hiệu suất và năng suất chưa từng có.

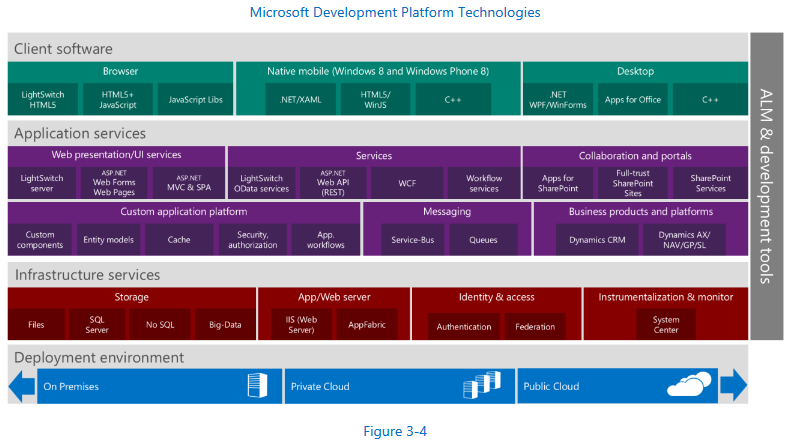
.NET cũng được xây dựng với những emerging patterns. Tại Diễn đàn 2000, Bill Gates đã nói rằng mục tiêu của .NET là "vượt xa thế giới, các trang web độc lập hiện nay trên Internet với các thành phần có thể hoán đổi cho nhau, nơi mà các thiết bị và dịch vụ có thể được lắp ráp thành những trải nghiệm gắn kết với người sử dụng". Viễn cảnh ban đầu cho .NET rất phù hợp với bối cảnh phát triển hiện nay, bao gồm trải nghiệm cross-device, service-powered.

Việc kích hoạt các trải nghiệm đa thiết bị được trao quyền bởi các dịch vụ là một thuộc tính quan trọng cho .NET ngay từ đầu. .NET đã tiếp tục phát triển kể từ đó, cung cấp các kinh nghiệm tốt nhất cho việc phát triển các nhu cầu mới của các ứng dụng:

* Về phía server side, .NET cung cấp một nền tảng chung cho các nhà phát triển để nhắm mục tiêu các dịch vụ chạy tại chỗ hoặc trong đám mây. Sự kết hợp chặt chẽ của nó với Windows Server và Windows Azure cho phép các ứng dụng dần dần được mở rộng lên đám mây, tận dụng tốt nhất từng nền tảng. Nhịp độ phát nhanh trong thư viện .NET Framework cũng cung cấp sự đổi mới liên tục đáp ứng nhu cầu mới của các ứng dụng dựa trên đám mây trong các lĩnh vực như lightweight services, real-time communications, mobile web applications, and authentication.
* Về phía client side, .NET cung cấp một trải nghiệm phát triển nhất quán, đầu tiên trên tất cả các thiết bị của Microsoft: trải nghiệm máy tính để bàn, ứng dụng Windows Phone và ứng dụng Windows Store (như trong Hình 3-3). Nó cho phép các nhà phát triển .NET tiếp tục phát triển các ứng dụng nền tảng trên máy tính để bàn và thêm những trải nghiệm mới thú vị, tất cả trong khi sử dụng các kỹ năng hiện có của họ và sử dụng lại mã giữa các thiết bị. Đối với các kịch bản mà phạm vi vượt xa các thiết bị của Microsoft, các giải pháp dựa trên trình duyệt HTML5 là tiêu chuẩn. .NET, kết hợp với Visual Studio, cung cấp một giải pháp hiện đại để tạo các ứng dụng web dựa trên tiêu chuẩn chạy trên các thiết bị khác nhau. Đối với các nhà phát triển muốn tạo ra những trải nghiệm trên bất kỳ thiết bị nào, các đối tác trong ngành Visual Studio cung cấp các giải pháp cho phép tái sử dụng C # và mã với các thiết bị không phải là Windows.

Tài liệu này truy cập tất các các lựa chọn cho phát triển .NET mà bạn có thể đưa ra các quyết định cho các kỹ năng của bạn và các yêu cầu chương trình khi bạn đưa các ứng dụng của bạn tiền về phía trước. Nó được cấu trúc để giải quyết hai mẫu ứng dụng

Hình 3-4 biểu diễn một sơ đồ toàn cầu của các công nghệ nền tảng phát triển của Microsoft. Các phần sắp tới sẽ được giới thiệu khi nào dùng các công nghệ này, tùy thuộc vào mô hình ứng dụng và ưu tiên



# Emerging application patterns

Như đã đề cập trước đây, các mẫu ứng dụng mới đang hình thành các ứng dụng của tương lai. Giờ đây, khách hàng và nhân viên yêu cầu các ứng dụng cung cấp trải nghiệm cá nhân nhiều hơn. Họ muốn ở liên tục kết nối với các dịch vụ họ cần.

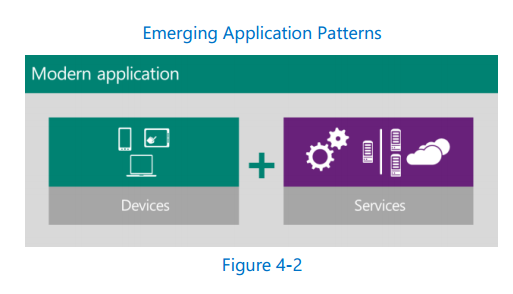
Phần này được cấu trúc bởi hai vấn đề chính cần được giải quyết khi phát triển loại ứng dụng mới này:

* Tạo trải nghiệm qua các thiết bị không đồng nhất.
* Tạo các dịch vụ tiêu chuẩn, nhẹ mà mở rộng qua đám mây

Emerging application patterns, theo nhiều cách, có thể so sánh với các "systems of engagement" (thuật ngữ được Geoffrey Moore đặt ra và cũng thường được Forrester sử dụng, như thể hiện trong hình 4-1), nhưng chúng được bổ sung - được hỗ trợ bởi cloud và phải dựa trên "systems of record"

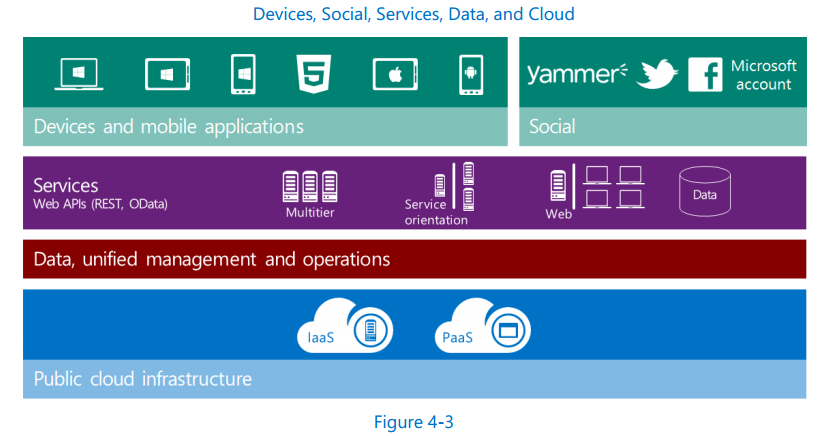
“Systems of engagement” không chỉ có nghĩa là "ứng dụng nhắm mục tiêu đến người tiêu dùng (applications targeting just consumers)". Trên thực tế, có nhiều kịch bản mới trong doanh nghiệp (ví dụ như các ứng dụng nội bộ có các yêu cầu về di động như bảng điều khiển), cùng với các mô hình nhắm mục tiêu **end customers** (như ngân hàng trực tuyến và thương mại điện tử), nơi khái niệm này là hoàn toàn hợp lệ.

Do đó, tầm nhìn của Microsoft về **emerging application patterns** bao gồm sự cần thiết của các dịch vụ liên tục và linh hoạt, ngoài nhu cầu di động và kết nối trực tiếp với người tiêu dùng, như trong Hình 4-2



Trong hầu hết các ứng dụng kinh doanh hiện đại, ứng dụng khách hàng sẽ cần một dịch vụ từ xa để làm việc với dữ liệu. Đây là nơi bạn cần định hướng dịch vụ dựa trên các tiêu chuẩn Internet cho các dịch vụ từ xa. Cuối cùng, các dịch vụ này sẽ được triển khai trong cơ sở hạ tầng và dịch vụ cloud, đây là những trụ cột **cho emerging patterns** liên quan đến môi trường triển khai.

Khi chúng tôi nghiên cứu thêm về **emerging application patterns**, bạn sẽ phải tính đến tất cả các thiết bị di động và mạng xã hội có thể có như một phương tiện để mở rộng các business applications của bạn sang các trường hợp sử dụng mới và để xây dựng các dịch vụ liên tục và vững chắc, Đáp ứng nhu cầu; Xem hình 4-3.



## **Devices**

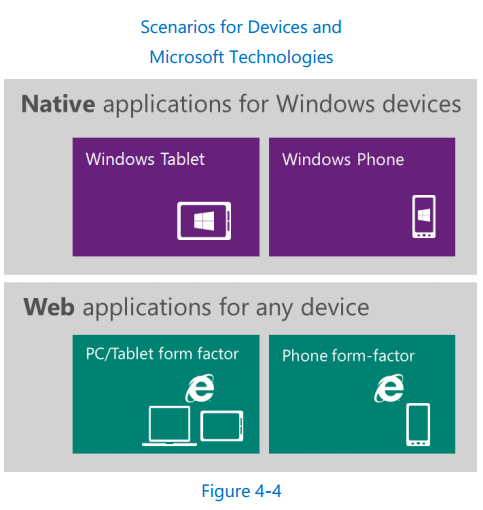
Khả năng cung cấp những trải nghiệm mới phù hợp với thiết bị là đặc tính chính cho các mẫu ứng dụng đang nổi lên. Việc lựa chọn công nghệ để tạo ra những ứng dụng đó có thể rất khó khăn và liên quan đến nhiều yếu tố, bao gồm:

* Kỹ năng và sở thích kỹ thuật trước đây của bạn.
* Khả năng tạo ra những trải nghiệm phù hợp kết hợp với các khả năng phần cứng cục bộ.
* Sự đa dạng của thiết bị mà ứng dụng của bạn sẽ nhắm mục tiêu.
* Công nghệ được sử dụng bởi các ứng dụng hiện tại của bạn cần di chuyển hoặc mở rộng tới thiết bị.

Hai lựa chọn thay thế thường được thiết lập trong ngành công nghiệp dựa trên những cách tiếp cận rất khác nhau:

* **Native applications**, có thể thu được nhiều nhất từ mọi thiết bị nhưng yêu cầu kỹ năng và mã độc đáo cho mỗi nền tảng.
* **Web applications**, có thể được tạo ra với một tập hợp các kỹ năng và mã thông dụng nhưng không thể cung cấp trải nghiệm phù hợp cho từng thiết bị

Nền tảng Microsoft hỗ trợ đầy đủ cả hai phương pháp tiếp cận (xem hình 4-4), nhưng nó làm giảm bất lợi của mỗi một cách đáng kể. Thứ nhất, các thiết bị Windows không thực thi một mô hình phát triển độc đáo. Bạn có thể sử dụng công nghệ có ý nghĩa nhất cho kỹ năng và ứng dụng hiện tại của bạn. Bằng cách đưa tích hợp thiết bị first-class sang .NET, HTML / JavaScript và C ++, bạn có thể đưa ra quyết định phù hợp nhất với nhu cầu của bạn mà không ảnh hưởng đến trải nghiệm. Thứ hai, .NET và Visual Studio đơn giản hóa việc tạo ra các ứng dụng web có thể chạy trên bất kỳ thiết bị nào. ASP.NET hoàn toàn đáp ứng các tiêu chuẩn hiện đại và kết hợp với những đổi mới độc đáo mới nhất trong Visual Studio cho phép tạo ra một loạt các ứng dụng web mới tận dụng được các trình duyệt hiện đại trên các thiết bị.



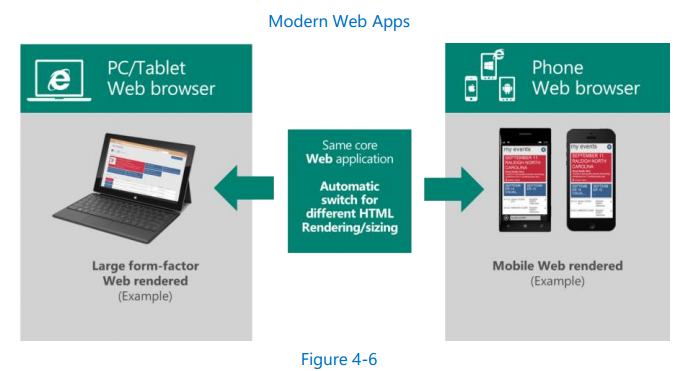
**Native applications for Windows devices**

**Native applications** là một ứng dụng chạy trên thiết bị của khách hàng và tận dụng các tính năng cụ thể của thiết bị đó để cung cấp trải nghiệm hấp dẫn nhất cho khách hàng. Như đã giải thích trước, nền tảng Windows mở rộng khái niệm này sang các công nghệ vượt xa C ++, mở rộng đáng kể khả năng sử dụng lại mã và kỹ năng của bạn để nhắm mục tiêu các yếu tố hình thức mới.

Bảng 4-1 và Bảng 4-2 giải thích sự khác biệt giữa các công nghệ này và khi chúng phù hợp để sử dụng, tùy thuộc vào mức độ ưu tiên ứng dụng của bạn và ngữ cảnh cụ thể.

**Web applications for any device**

Microsoft cung cấp các công cụ và công nghệ tốt nhất để tạo các ứng dụng HTML5 dựa trên trình duyệt chạy trên bất kỳ thiết bị nào. Bởi vì Windows cũng hỗ trợ HTML5 như một công nghệ bản địa, bạn có thể nhắm mục tiêu nhiều thiết bị bằng ứng dụng web HTML5, sau đó sử dụng lại và tối ưu hóa mã đó cho ứng dụng Windows Store gốc.

Việc chuyển đổi tự động cho việc hiển thị hoặc định dạng HTML khác nhau có thể đạt được thông qua các cơ chế khác nhau được cung cấp bởi ASP.NET, JavaScript, và LightSwitch, cũng như các thư viện như Modernizr, sẽ phát hiện trình duyệt và điều chỉnh mã HTML / JavaScript của bạn, như được hiển thị bên dưới Trong hình 4-6.

Cách tiếp cận này đặc biệt thuận lợi khi nhắm mục tiêu các hệ điều hành khác nhau (như Windows, iOS và Android). Sau khi đạt được khả năng tương thích với các trình duyệt web phổ biến nhất, ứng dụng của bạn sẽ được cross-platform tương thích với các hệ điều hành di động lớn.

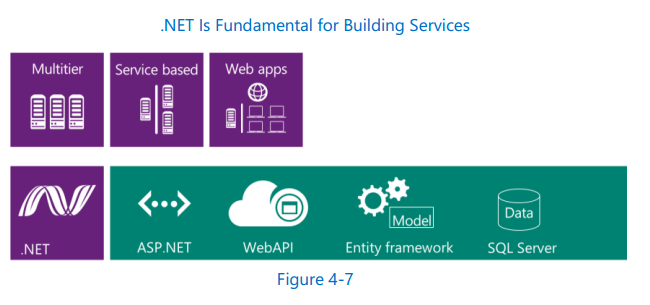
Bảng 4-3 liệt kê các công nghệ phát triển web có sẵn và khuyến cáo sử dụng mỗi công nghệ, tùy thuộc vào mức độ ưu tiên ứng dụng và bối cảnh của bạn

## **Services**

Quá trình nhắm mục tiêu nhiều thiết bị bắt đầu từ phía sau. Các ứng dụng cần để lộ các dịch vụ có thể được tiêu thụ trên bất kỳ thiết bị nào và thu nhỏ cho Internet.

Dịch vụ Internet có nghĩa là có tính sẵn sàng cao và liên tục với các dịch vụ kết thúc cho phép các ứng dụng hiện đại hoạt động được. Họ cũng phải nhanh nhẹn và có sự tiến triển liên tục, trơn tru, gắn bó với vận tốc thay đổi kinh doanh.

Trong bối cảnh này, “ **.NET ecosystem**” và **frameworks** là rất cần thiết khi xây dựng các dịch vụ.

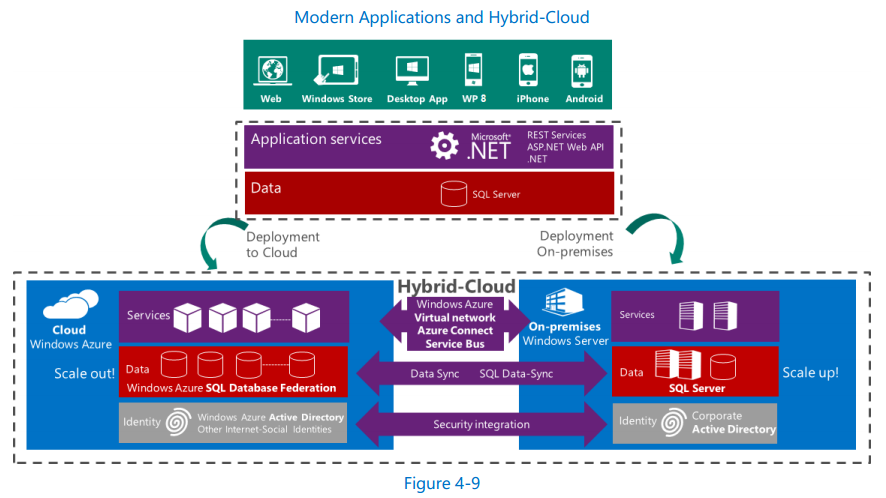
Như thể hiện trong hình 4-7, .NET Framework hỗ trợ đầy đủ các cách tiếp cận để xây dựng các ứng dụng: Các phương pháp tiếp cận ASP.NET để phát triển web (Single-page application [SPA], Model-view-controller [MVC], and WebForms) ; ASP.NET Web API để xây dựng các dịch vụ HTTP / REST; Và sử dụng entity framework để truy cập dữ liệu trong cơ sở dữ liệu quan hệ. Hầu hết sự phát triển phía máy chủ thường dựa trên công nghệ .NET.

## **Cloud and hybrid-cloud**

Các ứng dụng kinh doanh hiện đại thường hỗ trợ nhiều người dùng Internet (chẳng hạn như end customer và đối tác), do đó việc duy trì các dịch vụ kết thúc trong các trung tâm dữ liệu nội bộ của công ty bạn có thể không hợp lý. Vì lý do này, các dịch vụ có thể sẽ được triển khai trong đám mây. Các dịch vụ cũng được hưởng lợi từ các tính năng cốt lõi của đám mây, chẳng hạn như tính linh hoạt và thiết lập sản xuất nhanh và hiệu quả về chi phí.

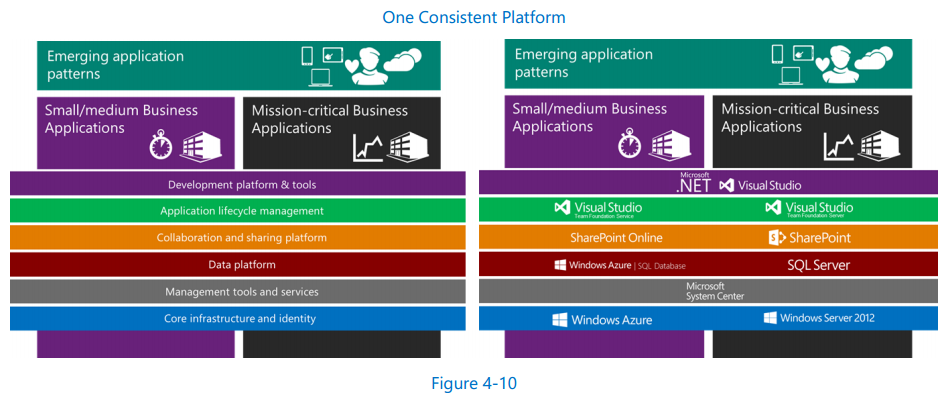
Ngoài ra, nhiều ứng dụng kinh doanh hiện đại là kết quả của những ý tưởng mới, các kênh mới và những cơ hội mới. Bạn không bao giờ biết nếu bạn sẽ thu hút được một số lượng đáng kể người dùng mới, do đó yêu cầu nhiều tài nguyên hơn những gì có sẵn trong trung tâm dữ liệu nội bộ của bạn. Đám mây là nhanh nhẹn và có thể mở rộng, vì vậy bạn có thể phát triển các khái niệm mới và nhanh chóng chuyển chúng sang sản xuất mà không cần đầu tư hoặc thiết lập phần cứng. Do đó, đám mây linh hoạt, như Windows Azure, là lựa chọn tốt nhất cho các ứng dụng kinh doanh hiện đại.

Trong một hệ sinh thái đa dạng, nơi các ứng dụng kinh doanh sống trong các môi trường khác nhau (cả trong đám mây hay on-premises), có nhiều nhu cầu mới cần đáp ứng. Bởi vì bạn cũng đang mở rộng các ứng dụng cơ bản đang on-premises, bạn cần một liên kết giữa các trung tâm dữ liệu cá nhân và đám mây, và bạn cần kết nối các thế giới này một cách an toàn thông qua một phương pháp **hybrid -cloud**, như trong hình 4-9 .



Nền tảng điện toán đám mây của Microsoft cung cấp kiến trúc, công nghệ và sản phẩm đối xứng để hỗ trợ các cơ sở hạ tầng trên đám mây và tại chỗ và cung cấp các dịch vụ chung để quản lý các ứng dụng trải dài trên hai cơ sở hạ tầng trong môi trường lai. Nền tảng này cho phép bạn di chuyển và xây dựng ứng dụng của bạn một cách dễ dàng và dần dần.

Như thể hiện trong hình 4-10, Microsoft cung cấp một nền tảng thống nhất, cho dù ứng dụng của bạn đang nhắm mục tiêu vào đám mây hay cơ sở hạ tầng on-premises. Đối với thực tiễn phát triển, cách tốt nhất là sử dụng nền tảng phát triển tương tự (ví dụ: Visual Studio hoặc .NET) cho cả môi trường đám mây và on-premises. Tương tự, việc có một hệ thống duy nhất để kiểm soát và quản lý cơ sở hạ tầng (như Trung tâm Hệ thống) cho bất kỳ hệ thống tại chỗ hoặc đám mây là cơ bản cho quản trị CNTT hiệu quả.

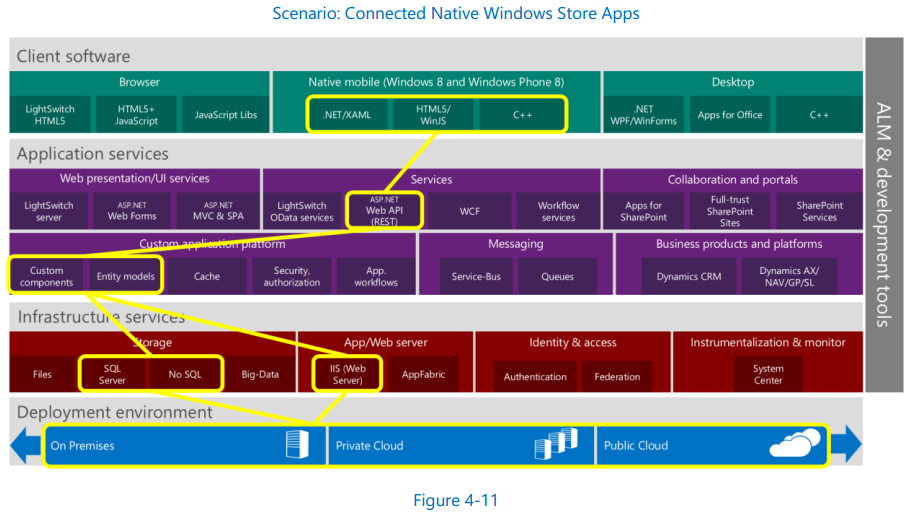


## **End-to-end scenarios in emerging application patterns**

***Scenario: Connected Native Windows Store Applications***

Sử dụng bản đồ công nghệ được giới thiệu trong Phần 3, "Overview"

Hình 4-11 cho thấy một kịch bản điển hình và mô hình kết nối công nghệ của nó cho một ứng dụng native Windows (bao gồm cả các tablet và mobile phone).



Kịch bản này sẽ tương tự nhau cho dù ứng dụng khách hàng giao diện người dùng của chúng tôi là native Windows Store app (ví dụ: Windows Tablet) hay native Windows Phone app. Tùy thuộc vào khả năng phát triển và yêu cầu ứng dụng của bạn và các mức độ ưu tiên, mà chúng tôi có thể lựa chọn giữa. / XAML, HTML5 / WinJS hoặc C ++, mặc dù chỉ có .NET và C ++ bao gồm Windows Store và phát triển ứng dụng Windows Phone native.

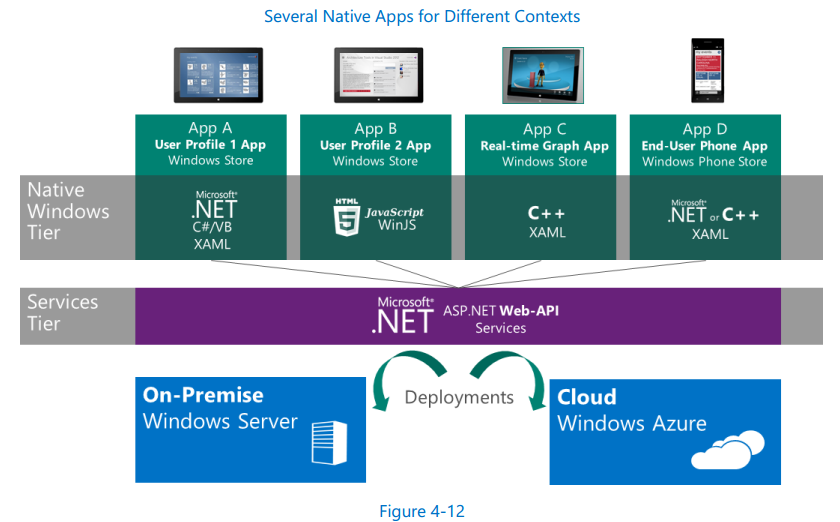
Đối với việc phát triển các dịch vụ tùy chỉnh nằm ở phần cuối của các ứng dụng hiện đại, công nghệ được đề nghị cho kịch bản này là ASP.NET Web API. Điều này cung cấp mức độ linh hoạt cao trong khi có một framework và hiệu suất cho các dịch vụ Internet, thường là nhóm các phương pháp như REST và OData hoặc JSON.

Sau đó, các dịch vụ Web API của ASP.NET sẽ bao quanh logic tầng trung gian dưới dạng các thư viện lớp tùy chỉnh và các mô hình thực thể. (Sử dụng entity framework nếu truy cập cơ sở dữ liệu quan hệ hoặc bất kỳ API nào khác nếu truy cập vào các nguồn dữ liệu NoSQL)...

Cuối cùng, chúng ta có thể triển khai các dịch vụ và các thành phần máy chủ của chúng ta trong bất kỳ môi trường triển khai nào, mặc dù trong các emerging application patterns, chúng ta sẽ thường triển khai nó vào một public cloud như Windows Azure.

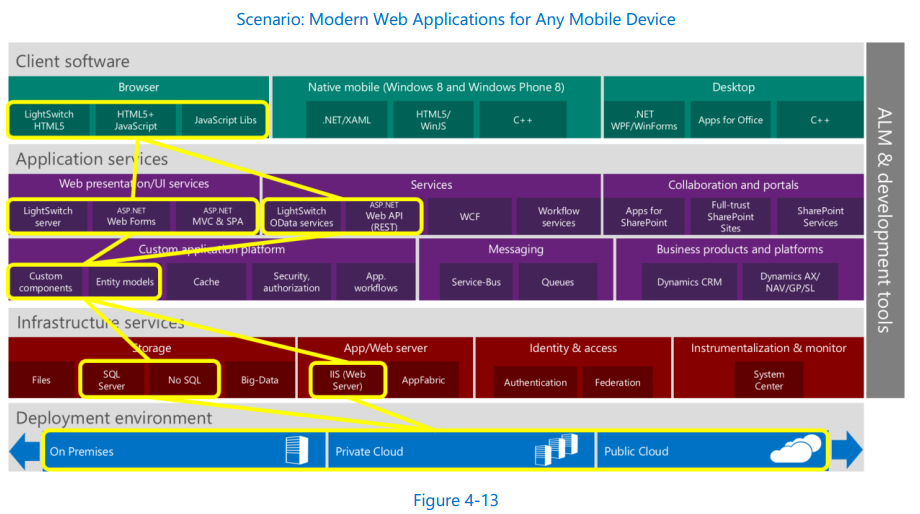
Như đã đề cập trước đây, một đặc trưng điển hình của các ứng dụng hiện đại là có các ứng dụng khác nhau cho các kịch bản ngữ cảnh / cá nhân khác nhau (như trong hình 4-12). Trong hầu hết các trường hợp, các tình huống này sẽ có những ưu tiên và yêu cầu khác nhau, dẫn đến quyết định của công nghệ. Ví dụ: khi phát triển ứng dụng Windows Store, đối với một tình huống cụ thể gần với nguồn cấp dữ liệu Internet và API mạng xã hội hoặc thậm chí khi bạn muốn sử dụng lại kỹ năng và mã JavaScript, phát triển HTML5 / WinJS sẽ là sự lựa chọn tốt nhất. Trong các trường hợp khác, nếu bạn muốn sử dụng kỹ năng C # hoặc XAML của bạn trong khi có hiệu suất tốt, bạn có thể sử dụng phương pháp tiếp cận NET / XAML. Và đôi khi có hiệu suất đồ họa tốt nhất có thể là một ưu tiên, trong trường hợp C + + sẽ là sự lựa chọn đúng.

Phát triển cho Windows Phone có các đề xuất tương tự vào tài khoản, cho dù bạn chọn .NET hay C ++.

Một điểm quan trọng khác là các dịch vụ back-end có thể giống nhau cho - và được tái sử dụng bởi tất cả các client /scenarios khác nhau mà bạn chọn. Ít nhất, bạn sẽ chia sẻ các công nghệ cấp cùng và máy chủ. Trong các trường hợp khác, bạn có thể có các mô hình dữ liệu, dịch vụ và hệ thống con khác nhau được tối ưu hóa cho từng ứng dụng khách hàng.

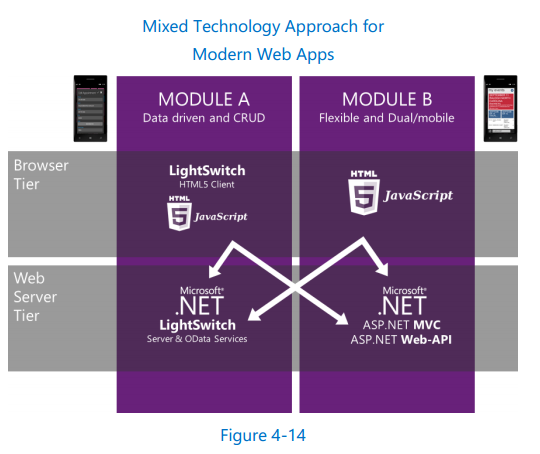
***Scenario: Modern Web Applications for Any Mobile Device (Tablets and Phone)***

Hình 4-13 cho thấy kịch bản ứng dụng web hiện đại điển hình và mô hình kết nối công nghệ khi nhắm mục tiêu bất kỳ yếu tố hình thức và bất kỳ nền tảng khách hàng nào (Windows hoặc không phải Microsoft).

Để phát triển ứng dụng web hiện đại hỗ trợ các thiết bị di động, các công nghệ được đề xuất là ASP.NET MVC với hỗ trợ HTML5 để phát triển web linh hoạt và toàn diện và khách hàng HTML5 cho các dự án LightSwitch khi giải quyết các kịch bản dữ liệu đơn giản.

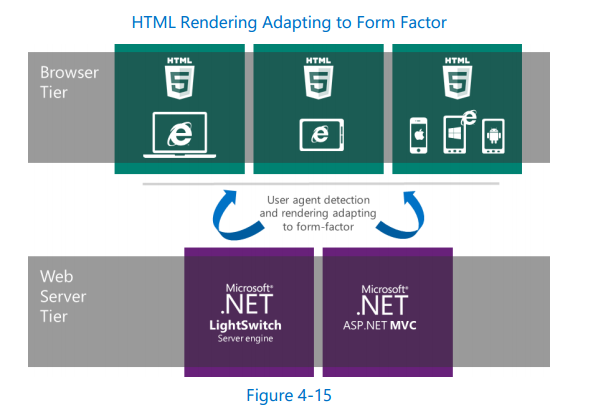
**Simplified sample architecture for modern web applications targeting any web browser and operating system platform**

Tùy thuộc vào yêu cầu và mức độ ưu tiên của ứng dụng web, bạn sẽ sử dụng các cách tiếp cận khác nhau. Đối với các ứng dụng có tính ứng dụng cao, hãy sử dụng LightSwitch, trong khi các kịch bản phức tạp hơn nên sử dụng các công nghệ hạt tốt hơn như ASP.NET MVC và HTML5 / JS.

Bạn có thể có nhiều modules với các ưu tiên và đặc điểm khác nhau (mạng dữ liệu so với các trang web phức tạp hơn) trong cùng một ứng dụng. Trong những trường hợp như vậy, có thể bạn sẽ áp dụng một cách tiếp cận hỗn hợp, như minh họa trong Hình 4-14.

Đối với trường hợp này, vì bạn đang sử dụng các công nghệ web chuẩn, bạn có thể nhắm mục tiêu bất kỳ thiết bị và bất kỳ hệ điều hành nào, không chỉ là một tập hợp các thiết bị di động.

Nếu ứng dụng đang nhắm mục tiêu đến các yếu tố hình thức khác nhau (điện thoại và máy tính bảng), bạn nên xem xét nếu mỗi yếu tố hình thức sẽ có lợi từ bố cục hoặc chế độ xem khác.

Trong hình 4-15, các dịch vụ render máy chủ (cho dù là ASP.NET MVC hay LightSwitch) có khả năng hiển thị mã HTML để hỗ trợ các giải pháp thay thế nhiều thiết bị. HTML có thể được hiển thị cho mỗi trình duyệt dựa trên đại lý người dùng web. Trong trường hợp của ASP.NET MVC, các tệp khác nhau phải được tạo cho mỗi tác nhân người dùng, vì đó là trình tạo cấp thấp hơn. Với LightSwitch, các render khác nhau được tạo ra tự động, làm đơn giản hoá đáng kể việc phát triển ứng dụng đa kênh.