

LIVE! 360
TECH EVENTS WITH PERSPECTIVE

Modern App Architecture

Brent Edwards
Principal Lead Consultant
Magenic

Level: Intermediate

The Ultimate Education Destination
Orlando 2016
ROYAL PACIFIC RESORT AT UNIVERSAL
DECEMBER 5-9

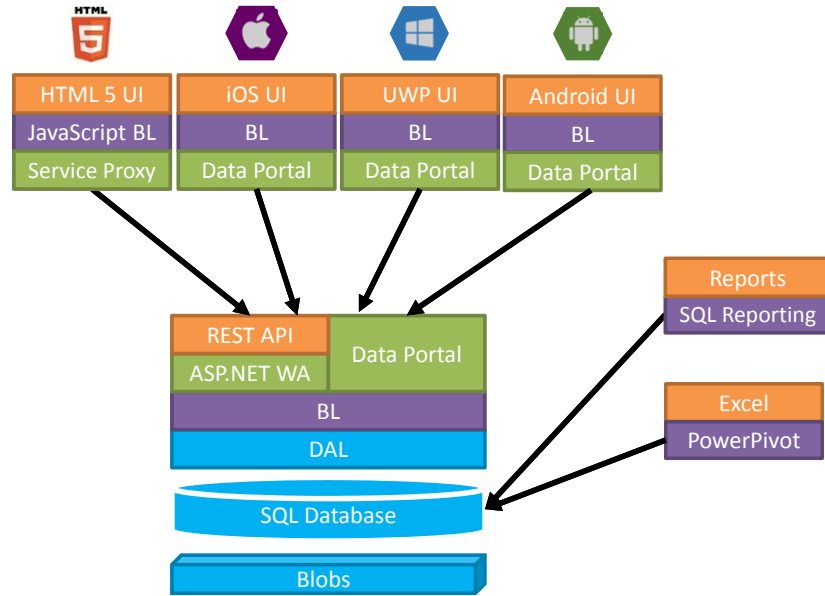
Visual Studio **LIVE!** | SQL Server **LIVE!** | **TECHMENTOR** | Office & SharePoint **LIVE!** | ModernApps **LIVE!** | **NEW!** **APPEV TRENDS**

BrentE@magenic.com
@brentledwards
<http://www.brentedwards.net>
<https://github.com/brentedwards>

BRENT EDWARDS



MyVote App Architecture



PUTTING IT TOGETHER

MyVote Development Goals

Efficient use of distributed team

- Each member focuses on strengths

Well designed architecture

- Works across many platforms

Maintainability

- Minimize duplication of code
- Maximize re-use of code



THE FOUNDATION



Layered Architecture

Organize
your...

Application code

Test plans and execution

Development team

Manage
complexity

Related functions go in a layer

Layers are isolated

Layers communicate



Logical Layers

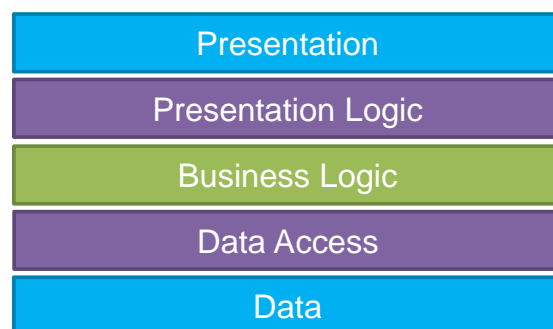
- Breaks application into logical layers
 - Layers may all run on machine
 - Layers may run on separate machines
- Provides
 - Maintainability
 - Readability
 - Flexibility
 - Reuse
 - Lower development/maintenance costs



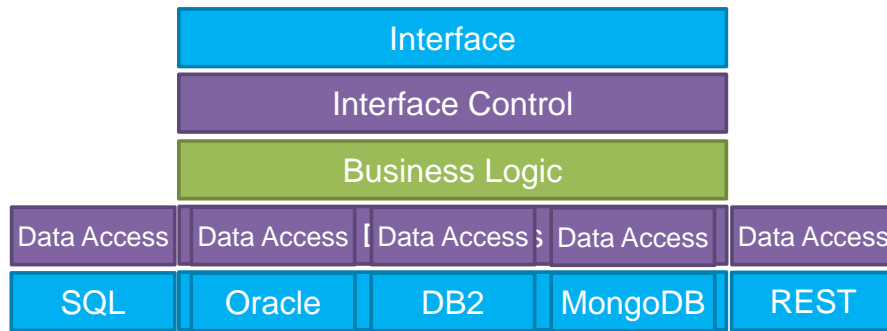
Anatomy of a Modern App



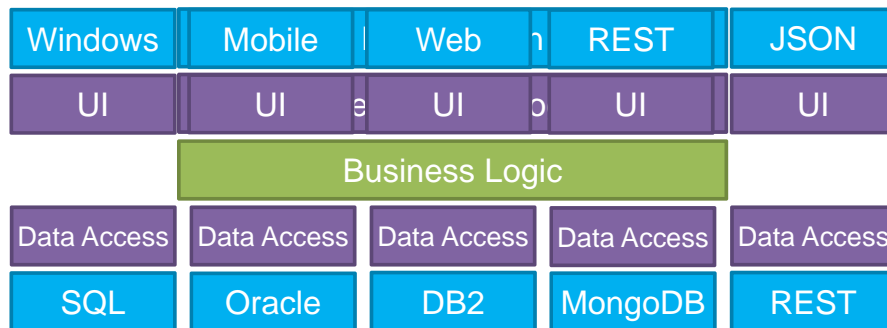
N-Layer



Multiple Data Sources



Multiple Presentations

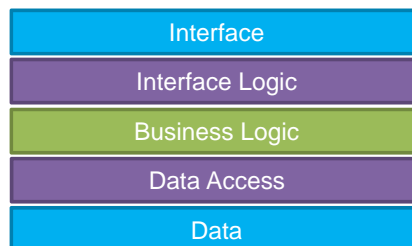


Physical Tiers

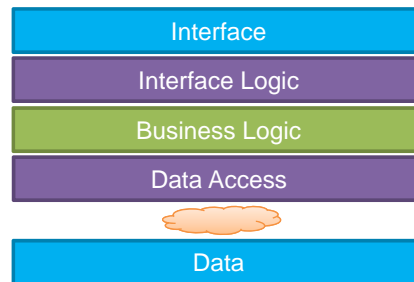
- Deploy logical layers to physical tiers
- Trade off between
 - Performance
 - Scalability
 - Fault tolerance
 - Security



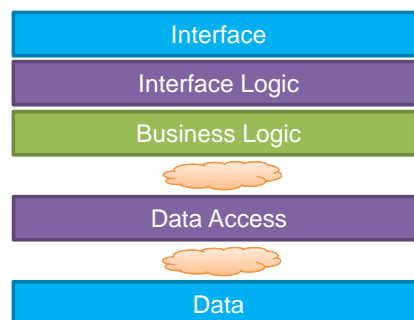
N-Layer App Architecture



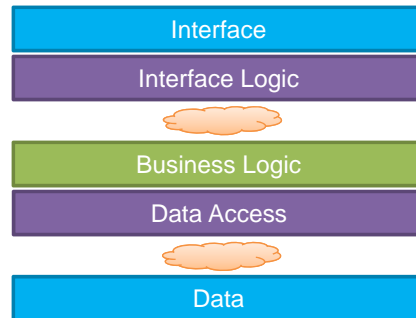
2-Tier App Architecture



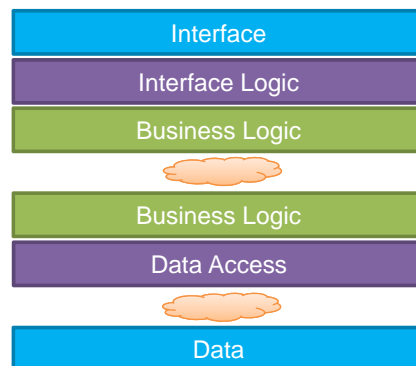
3-Tier App Architecture (client logic)



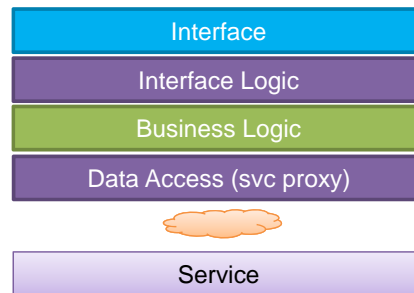
3-Tier App Architecture (server logic)



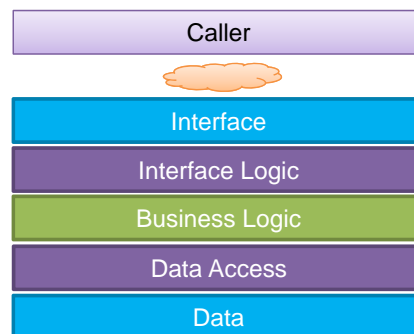
3-Tier App Architecture (shared logic)



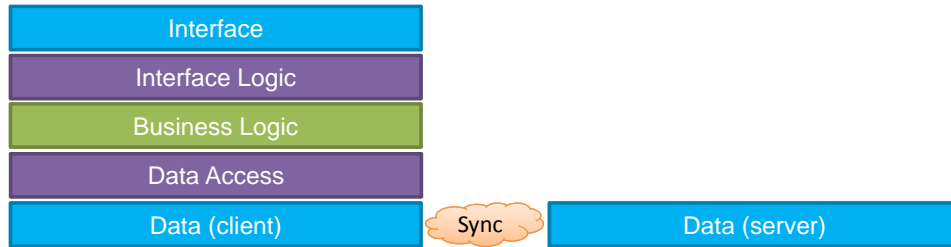
Edge App Architecture



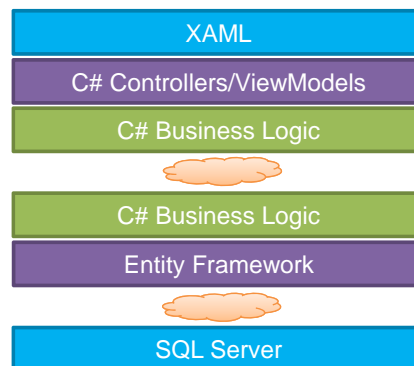
Service Architecture



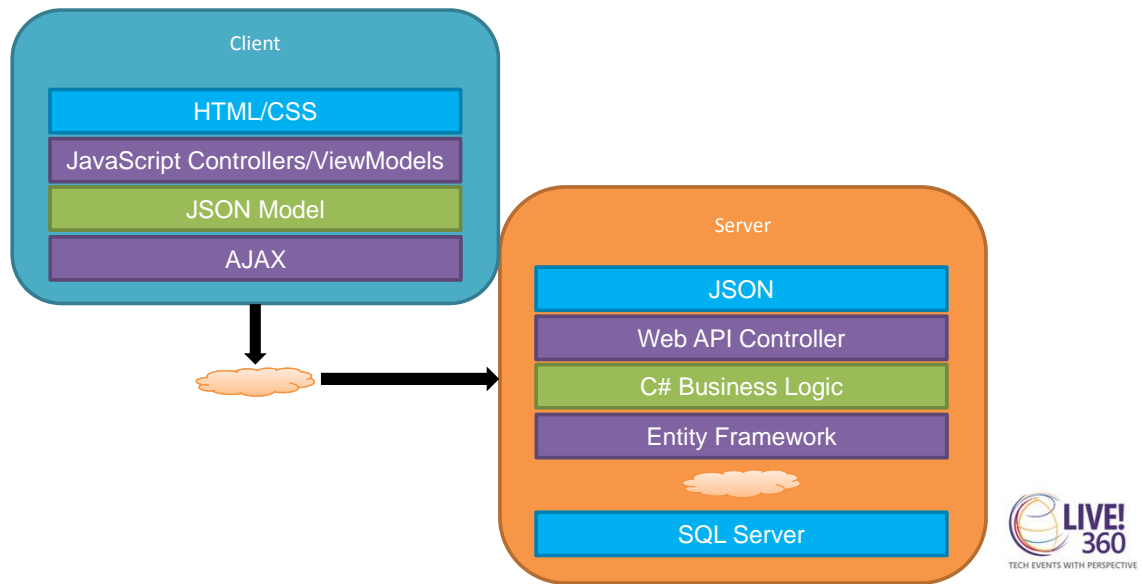
Disconnected Edge App Architecture



MyVote Windows, iOS, Android Apps



MyVote HTML 5 App



MyVote: Other Concerns

- Authentication
- Authorization
- Poll Image Storage

WHAT DOES IT ALL MEAN?



The Multi-Platform Conundrum

- Code re-use is ideal
 - But only an option for .NET platforms
- What about the rest?
 - Must duplicate effort



Combat Code Duplication

- Maintain all code for all platforms?
- Generate platform-specific code based on C# code?
 - How?
- Complete code generation?
 - How?
 - Some sort of meta language?
 - What tool do you use?
- Use Mono/Xamarin?



Can you afford multiple dev teams?

- If so...
 - Which platforms do you target?
 - How many should you target?
- If not...
 - Do you choose some platforms?
 - Which ones?
 - Do you choose a cross-platform solution?
 - Which one?



WHAT ABOUT VERSIONING?



Versioning

- With new versions come additional maintenance
- Client and Server can get out of sync
- Likely leads to maintaining multiple versions



Versioning Options

- Maintain different REST API Urls
 - As clients update, they get the new urls
 - Double (or more!) maintenance
 - Might be ok for short-term situations (days/weeks)
 - Long-term nightmare
- Make Smart Service
 - Parse doc and determine version
 - More work, but not multiple maintenance
 - Can be used for months or years



WHAT'S THE RIGHT ANSWER?



42



Architecture is hard!



Source: <http://blu.stb.s-msn.com//FE/A31AA56A3AD65F85FBA513BD4AA5.gif>



MYVOTE: AN EXAMPLE ARCHITECTURE



Technologies: How to Decide?

- Your Team
 - What are their strengths/weaknesses?
 - What are their interests?
- Costs
 - What costs are associated with each choice?
- Goals
 - Support multiple client platforms?
- Landscape
 - What are the most popular and supported options?
- Future
 - Where are technologies headed?

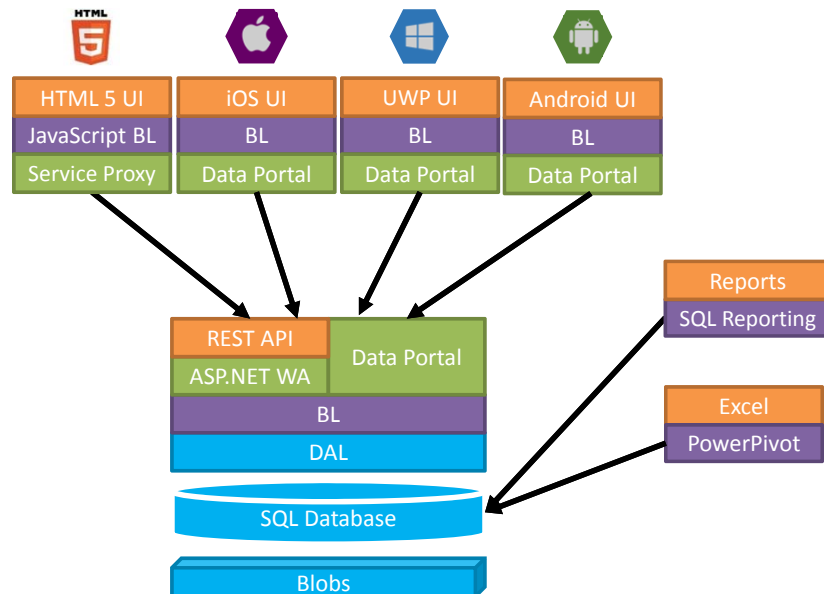


MyVote Team

- Strong background in
 - Microsoft
 - Web
 - Mobile
 - Xamarin



MyVote App Architecture



MYVOTE SERVER

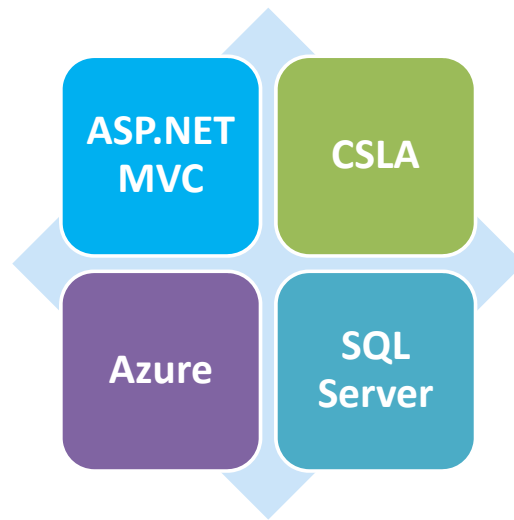


MyVote Server Needs

- Hosting for
 - Website
 - Web API
- SQL Server Database
- Blob Storage
- Business Rules and Validation



MyVote Server Technologies



Microsoft Azure

- Multi-faceted Cloud Solution
 - All our needs met under one roof
- Infinitely Scalable
- Dynamic Pricing



Azure SQL Server

- Scalable
- Data is Replicated
- SQL Server Data Tools (SSDT)
 - Visual Studio Database Project
 - Can check into source control
- Entity Framework for interaction



Azure Mobile Services

- Authentication
- OAuth
 - Microsoft
 - Facebook
 - Twitter
 - Google



Azure Blob Storage

- Elastic Scaling
- Access via REST API
- Store Poll Images as Blobs
- Give upload access on-demand
 - Generate Storage Access Security (SAS) Key



Azure Website

- ASP.NET MVC
- Web API
- Hosts
 - CSLA DataPortal
 - REST API
 - Web App



CSLA

- Cross-Platform Business Object Framework
- Open Source
 - <http://cslanet.com/>
- Support for Business Rules and Validation
- Same code runs on client and server
 - Kind of...
 - Same class files
 - “Server” code interacts with database
 - “Client” code can’t (and shouldn’t) interact with database



MYVOTE CLIENTS



MyVote Client Apps

- Goals
 - As many platforms as possible (for us)
 - Match platform look-and-feel
 - Maximize code re-use



MyVote Client Platforms



MyVote Client Apps

- Technology Options
 - Mobile Web
 - Fully Native
 - Cross-platform solution
 - Xamarin
 - Cordova



Tech Option: Mobile Web

- Pros
 - Already lots of experience with web
 - Can reuse all code
- Cons
 - No re-use of server's business logic
 - No native look-and-feel out of the box



Tech Option: Fully Native

- Pros
 - Native look-and-feel
 - Build apps as each platform intends
- Cons
 - No code re-use
 - This is a biggie



Tech Option: Cordova

- Pros
 - Cross-platform
 - Lots of code re-use
 - Builds on Web dev knowledge
 - Don't need Windows and Mac machines to build
- Cons
 - No re-use of server's business logic
 - Users can sometimes tell it's not a "native" app



Tech Option: Xamarin

- Pros
 - Native look-and-feel
 - Lots of code re-use
 - Can re-use server's business logic
- Cons
 - Can't re-use UI code
 - Requires expensive license
 - Need Windows and Mac machines to build



Tech Option: Xamarin.Forms

- Pros
 - Pros of Xamarin
 - Can re-use UI code
- Cons
 - Still requires expensive license
 - Still need Windows and Mac machines to build



MyVote Client Apps

- Our Technology Decision
 - Universal Windows Platform
 - Strong Windows dev background
 - Xamarin.Forms for Android/iOS
 - Strong Xamarin dev background
 - JavaScript for Web
 - Strong web dev background



Code Re-Use

- Universal Windows
 - Now same code on all Windows platforms
- Android/iOS
 - Most code shared
 - Except for UI
- Web
 - No code shared
 - Island of its own



MyVote Mobile Platforms Tech

- Universal Windows
 - With familiar .NET feel
- Xamarin.Forms
 - With familiar .NET and XAML feel
- MvvmCross
 - With support for all the .NET-base client tech
- CSLA
 - With support for all the .NET-base client tech



MyVote Web Platform Tech

- JavaScript with Typescript
 - Makes JS feel more like C#
- AngularJS
 - Very popular with tons of support and examples
- Responsive Design with Bootstrap
 - Also very popular with tons of support and examples



PARTING THOUGHTS



Parting Thoughts

- Leverage Layered Architecture
- Spread layers across tiers
- Be prepared for logic duplication
- Know your team's current and potential capabilities
- Remember to KISS
 - Keep It Simple



QUESTIONS?

