

PUG Challenge Americas

2013 – Westford, MA



Application Evolution Success Stories

Presented by: Dustin Grau & Chris Longo

BRAVEPOINT

About BravePoint, Inc.



- Founded in 1987
- Purchase in 1991 by Chesapeake Utilities Corporation (NYSE:CPK) and is a wholly-owned subsidiary company
- Headquartered in Atlanta, GA
- Currently ~100 employees and associates
- Services include a wide range of consulting, mentoring, training, and placement services
- Offering modernization strategies since 2009



- Success Stories
- How We Did It
- Demonstration



- Technology Integration
 - ErpXtend
 - Varnet
 - TirePro
 - Custom



- Sinclair Dental - Desktop (Adobe Flex)
- Precision Strip - Desktop (SmartClient)
- York Insurance - Desktop/Mobile
- Kauffman Tire - POS (Tablet)
- Sterling Jewelers - POS (Tablet)



- ProfitZoom Product
 - Complete rewrite of 2 applications
 - Web-based desktop (AJAX/RIA)
 - Mobile application (Android)
 - Leverages reusable logic for both
- Moneypenny Application
 - WebSpeed app, running 10+ years
 - Very procedural code (Progress v9)
 - Addon module for mobile access
 - Reused logic without modification

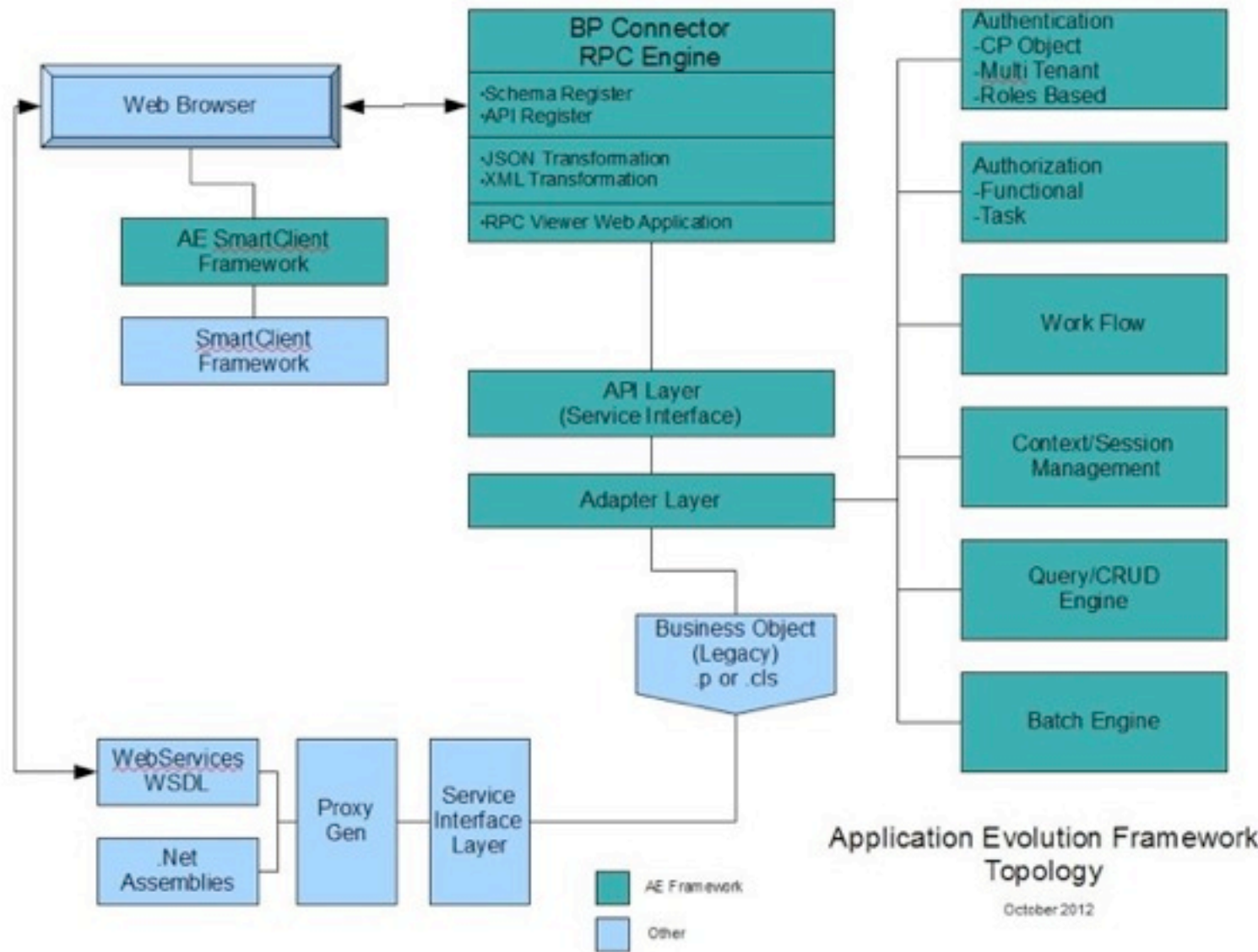


- Using the Application Evolution framework, of course!
 - Layered business logic (server side)
 - Centralized data layer (middleware)
 - Template based UI (client side)
- Core idea is Separation of Concerns (SoC)
 - CS design principle for separating a program into sections
 - Each distinct section addresses a separate concern
 - Section = module = layer, with a well-defined interface
 - Internal operations are small, atomic, and stackable



- Procedures
- Services (SOA)
- Model-View-Controller
- OSI Model (TCP/IP Network)
- Household Circuit Breaker

AE Overview



Application Evolution Framework Topology

October 2012



- Provides critical OERA framework features
 - Session and context management
 - Authentication and security
 - Messaging and error handling
- Layered architecture
 - API's: open to the outside world
 - Adapters: object-oriented structure
 - Interfaces: gateway to legacy code
- Lightweight and flexible approach
 - Does not require .NET or 3rd-party components



- Standardized gateway for data
 - JSON or XML
 - RPC or REST
- Uniform result and error handling
 - Removes burden on developer
- Supports HTTP/HTTPS
 - Any AJAX-based RIA
 - Most mobile devices



- Default demo uses SmartClient RIA
 - BP templates provide a starting point
 - Avoid being intrusive, allows customization
 - Inheritance structure supports extensibility
 - Provides standard for data binding with middleware
- Other frameworks are supported
 - Easy success with JQuery
 - Some frameworks require more work
- Android and iOS support JSON data



- UI design tools can speed development, to a point
 - Learning curve includes framework and tool use
- Rigid component structure provides standardization
 - Can result in lack of flexibility for custom components
- Can provide direct export to devices or environments
 - Consider ability to export code for source control
- For GUI environments, check if code can be modified
 - May be difficult to isolate problems or debug without access



- Design process is 1/3 of entire project timeline
 - Small changes early can avoid large re-writes later
- Tools such as iRise can create usable mockups
 - Allows for C-level and non-technical review
- Flowchart software creates detailed workflows
 - Leaves no ambiguity or gaps within logic
- Create basic layout with UI of choice
 - Simple wireframe with stubbed data calls

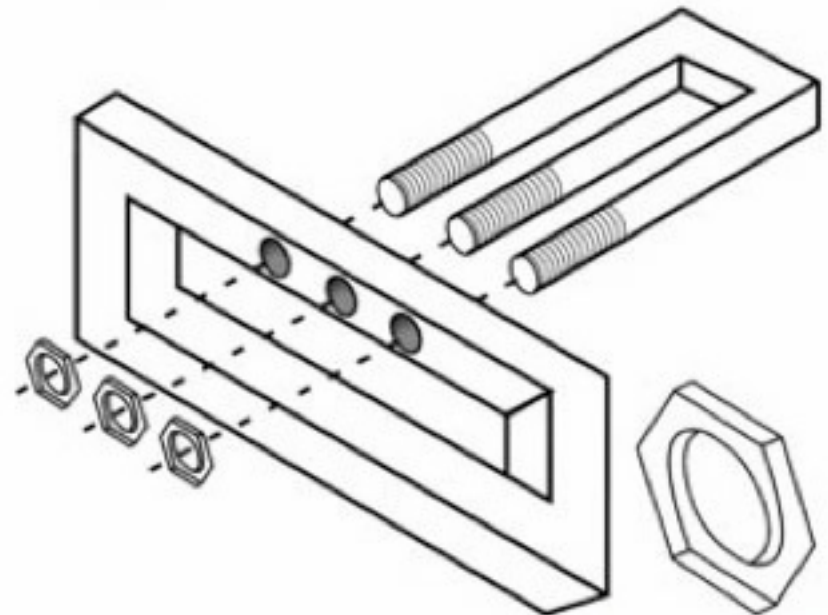


- GLOBAL variables
 - Handled in Interface layer(s)
- Forms and Frames
 - Manual separation of UI needed
- Data input requests (READKEY)
 - Must re-structure logic flow to accommodate
- Procedural code
 - Could be reusable, check error handling
- Poor/missing error handling
 - Centralized by data layer, use CATCH/THROW

DIY Parts List



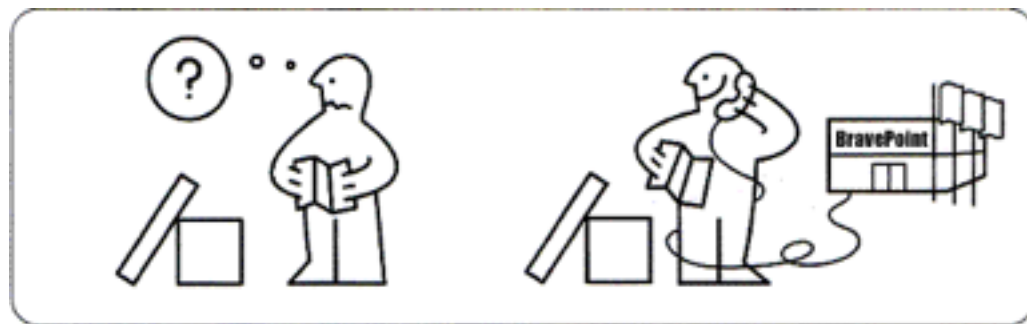
- 1x AJAX Framework (e.g. JQuery)
- 1x MyAPI.p with PROCEDURE getData
- 1x WebSpeed Broker
- 1x Progress.Json.ObjectModel (OE11+)
- ?x GET-SIGNATURE
- ?x CREATE CALL
 - ?x SET-PARAMETER
 - ?x GET-PARAMETER



Assembly Instructions



- Parse URL and JSON data (if any)
- Determine API to use and procedure to run
- Read signature of procedure and gather parameters
- Use CREATE CALL and SET-PARAMETER
- Dynamically invoke procedure and run logic
- Collect output values with GET-PARAMETER
- Prepare new JSON data for return
 - If error, create as appropriate object





- Eclipse (Developer Studio for OpenEdge)
- Apache ANT for cross-platform scripting
- Progress Compiler Tools (PCT) - Thanks Gilles!
- Subversion (Subclipse, TortoiseSVN)



- Moneypenny
 - Desktop
 - Mobile

BRAVEPOINT

Timesheet Entry for Dustin Grau
Core, Consultant

Total Hours: 31
Week Ending: 06/15/13
Mon, Jun 10, 2013

Current Entry:
Select a Job Number Here
Task: Select a Job Number First (From Above)
Descr: Timesheet Date: 06/10/13 End Entry on: (Optional)

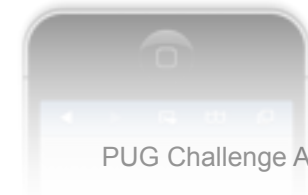
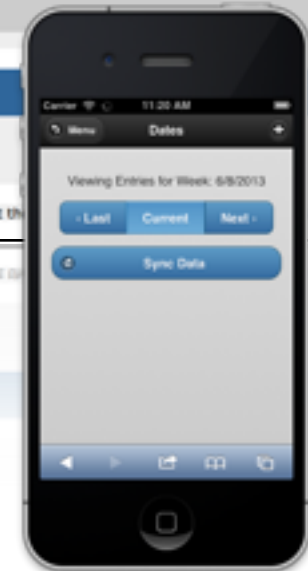
Accept Discard New

Entries for 06/10/13:
Job: 90120-0-0 - Employee Classrm. Education/Conferences
Task: 0-WP - Work in Progress (Default) 8.00 hrs
State: MA Status: In Progress (EGR)
Descr: PUG Challenge Americas

Other Entries for Week Ending 06/15/13:
Choose an Entry Below, then Click the Icon to Copy to the Currently Selected Day ->

Sun, 6/9	Mon, 6/10	Tue, 6/11	Wed, 6/12	Sat, 6/15
6 Hours in 2 Tasks for 2 Projects	8 Hours in 1 Task for 1 Project	8 Hours in 1 Task for 1 Project	9 Hours in 2 Tasks for 2 Projects	0 Hours in 0 Tasks for 0 Projects

Please view the Site FAQ or Knowledge Base if you need help. Or you can contact th



Thank You!



Questions?