

---

# Easy Web Service on iOS with Pico

-by William

<http://bulldog2011.github.com/>

---

---

# Problem Domain

- There is a gap between iOS device and traditional SOAP/XML based web service

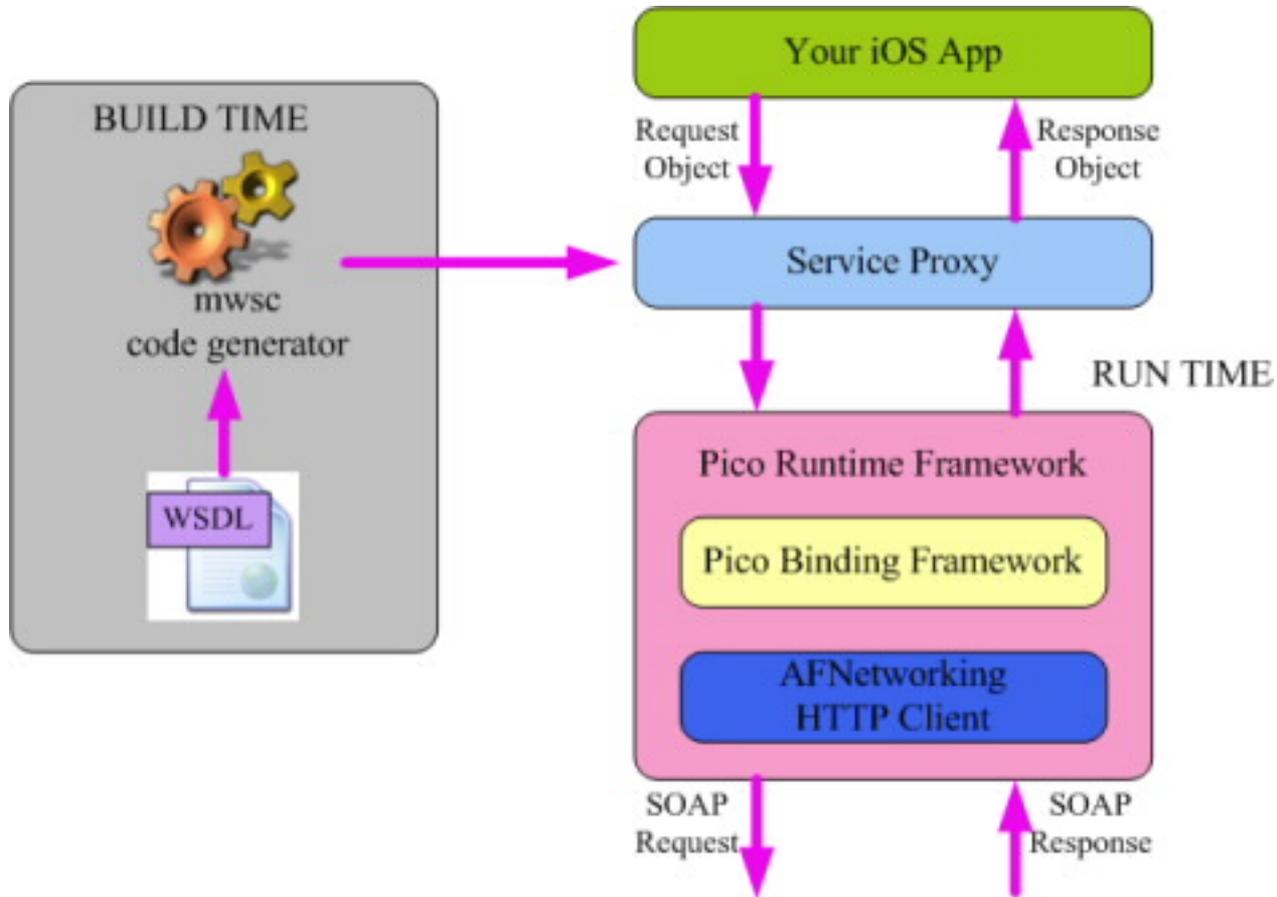


---

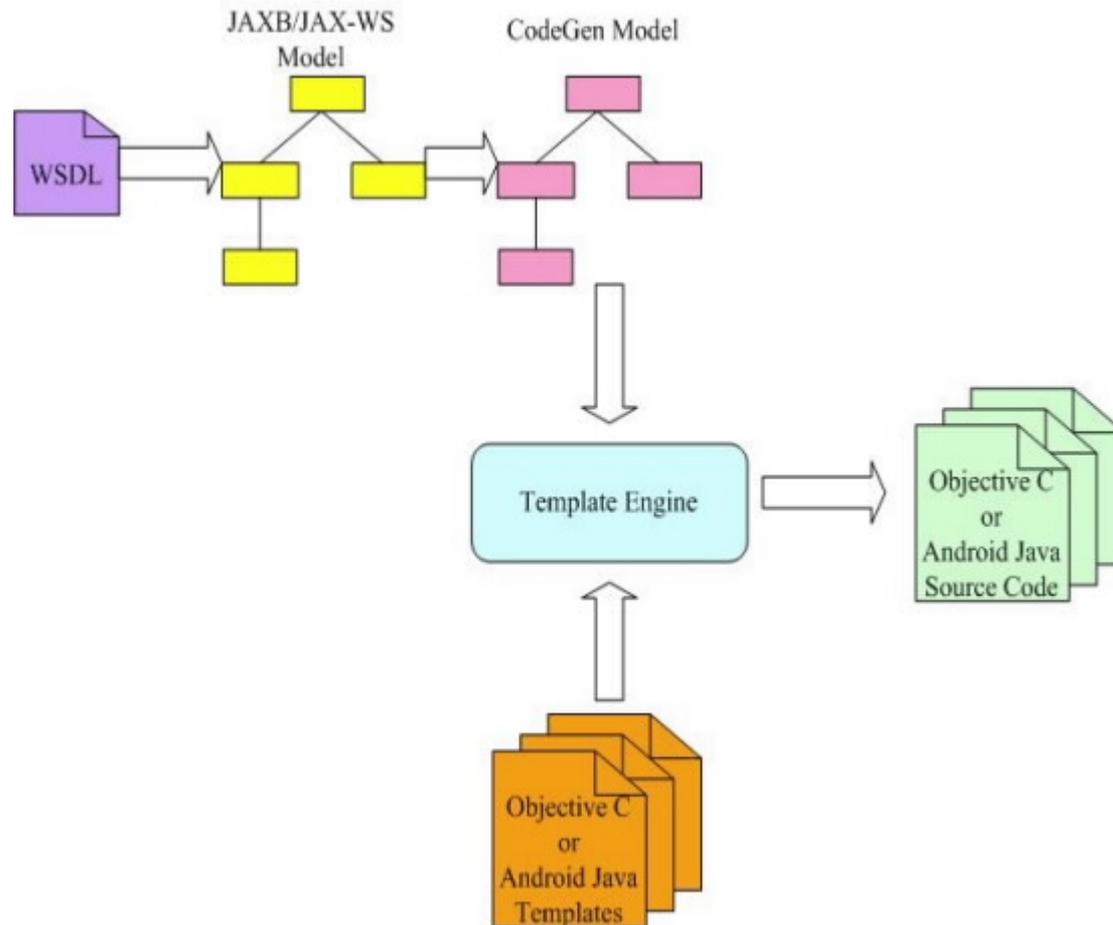
# Pico to Fill the Gap

- Pico is a light client-side web service framework tailored for iOS platform.
  - Feature Highlight:
    - Support WSDL driven development, auto-generate proxy from WSDL,
    - Support SOAP 1.1/1.2 and XML based Web service,
    - Automatic XML to Objective-C binding,
    - Built on popular and mature AFNetworking library,
    - Verified with Industrial grade Web Services like Amazon and eBay Web Services,
    - Asynchronous service invocation, flexible HTTP/SOAP header, timeout, encoding setting, logging, etc.
-

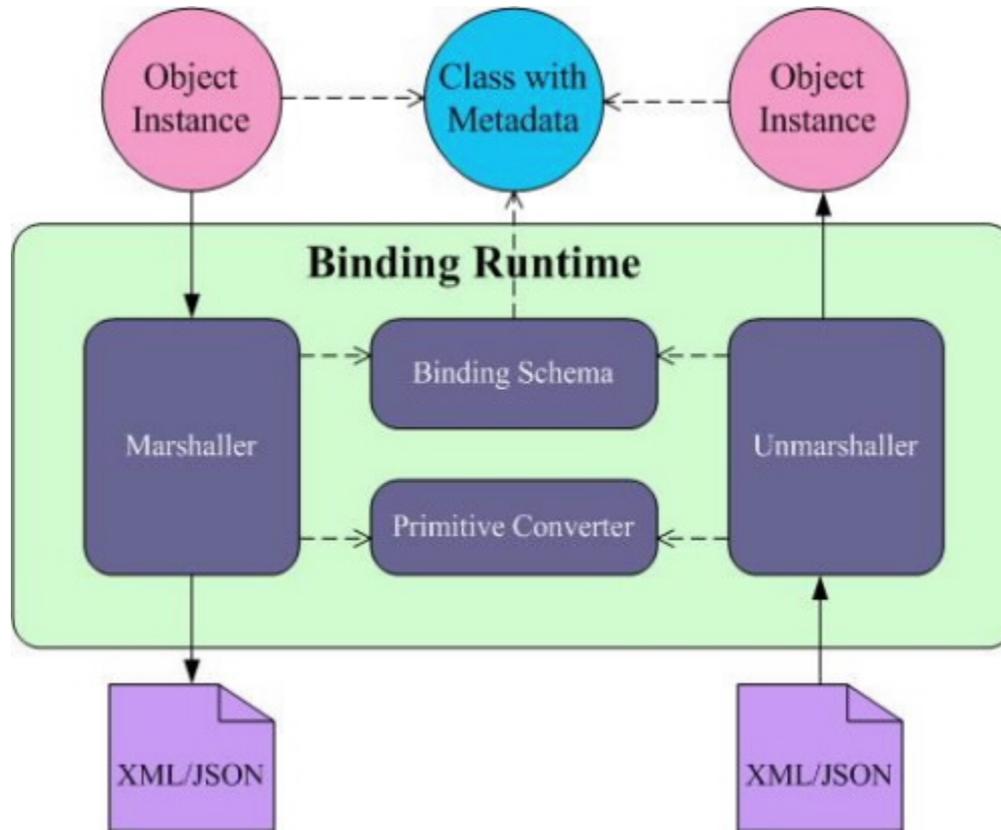
# The Big Picture



# Code Generation from WSDL



# Automatic XML<>Object Binding



---

# WSDL Driven Dev Flow on iOS

1. Generate Objective-C proxy from WSDL,
  2. Create new iOS project, add Pico runtime and generated proxy into the project,
  3. Implement application logic and UI, call proxy to invoke web service as needed.
-

# Simple Service Invocation Paradigm

- Invoke service with:
  1. Request object
  2. Success callback for success handling logic
  3. Failure callback for failure handling logic

```
/**  
 * Get Stock quote for a company Symbol  
 */  
-(void)getQuote:(GetQuote *) requestObject  
    success:(void (^)(GetQuoteResponse *responseObject))success  
    failure:(void (^)(NSError *error, id<PicoBindable> soapFault))failure;
```

# A Service Call Sample

```
// start progress activity
[self.view makeToastActivity];

// Get shared service client
StockQuoteServiceClient *client = [StockQuoteServiceClient sharedClient];
client.debug = YES; // enable request/response message logging

// Build request object
GetQuote *request = [[[GetQuote alloc] init] autorelease];
request.symbol = _symbolText.text;

// make API call and register callbacks
[client getQuote:request success:^(GetQuoteResponse *responseObject) {

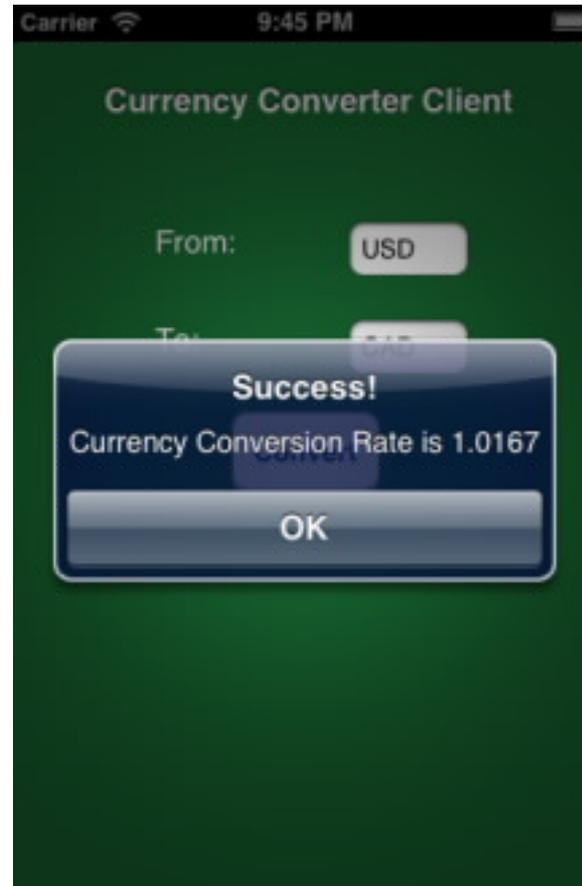
    // stop progress activity
    [self.view hideToastActivity];

    // show result
    _resultText.text = responseObject.getQuoteResult;
} failure:^(NSError *error, id<PicoBindable> soapFault) {

    // stop progress activity
    [self.view hideToastActivity];

    if (error) { // http or parsing error
        [self.view makeToast:[error localizedDescription] duration:3.0 position:@"center" title:@"Error"];
    } else if (soapFault) {
        SOAP11Fault *soap11Fault = (SOAP11Fault *)soapFault;
        [self.view makeToast:soap11Fault.faultstring duration:3.0 position:@"center" title:@"SOAP Fault"];
    }
}];
```

# Demo 1 – Currency Converter



# Demo 2 – Amazon Book Finder

Amazon Product Advertising  
API used:

1. ***itemSearch*** for book search
2. ***cartCreate*** to add chosen book into shopping cart



# Demo 3 – eBay Demo App

eBay Finding API used:

1. ***findItemsByKeywords*** for item search

eBay Shopping API used:

2. ***getSingleItem*** for item details

eBay Trading API used:

3. ***addToWatchList*** for adding item to watch list
4. ***getMyeBayBuying*** for watch list view



---

# Source, Samples and Tutorials

<https://github.com/bulldog2011/pico>

---