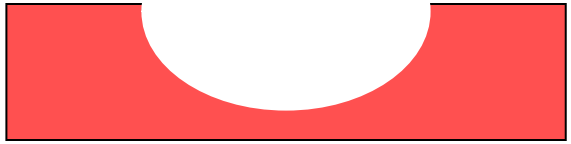
Easy Web Service on Android with Nano

-by William
http://bulldog2011.github.com/

Problem Domain

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

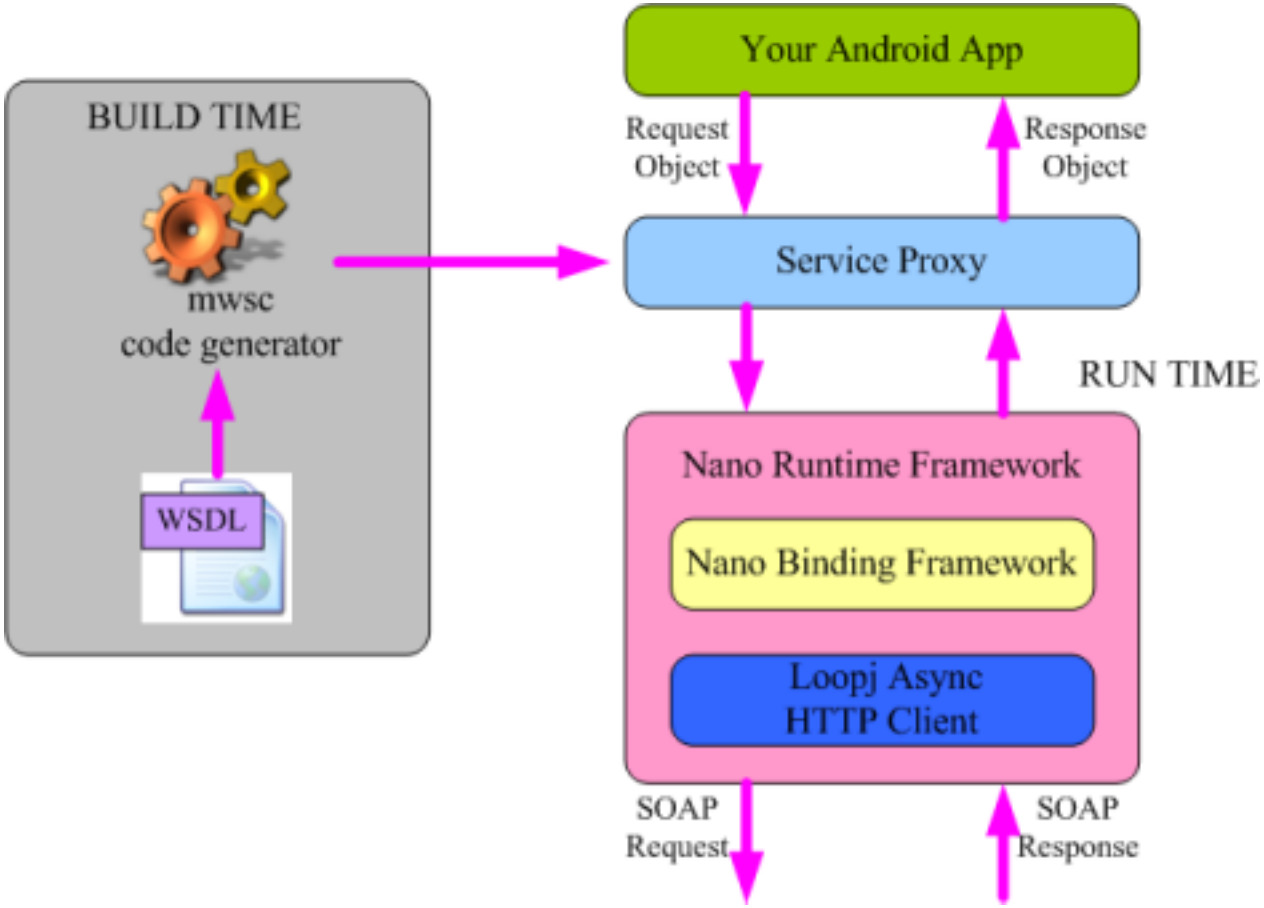


SOAP/XML
Web Service

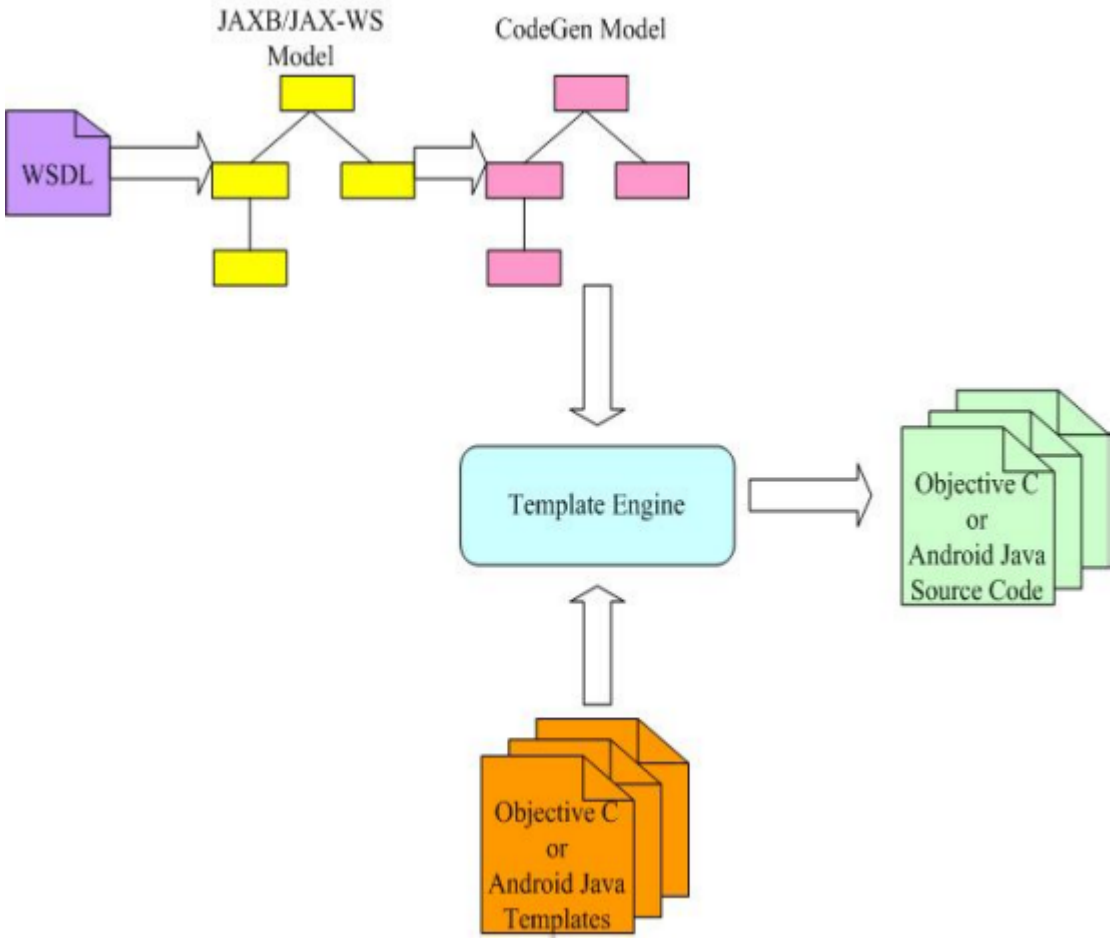
Nano to Fill the Gap

- Nano is a light client-side web service framework tailored for Android platform.
- Feature Highlight:
  - Support WSDL driven development, auto-generate strongly typed proxy from WSDL.
  - Support SOAP 1.1/1.2 and XML based Web service.
  - Automatic XML to Java binding, performance comparable to Android native XML parser.
  - Verified with industrial grade Web Services like Amazon and eBay Web Services.
  - Asynchronous service invocation, flexible HTTP/SOAP header, timeout, encoding setting, logging, etc.
  - Can be used as a standalone XML and JSON binding framework.

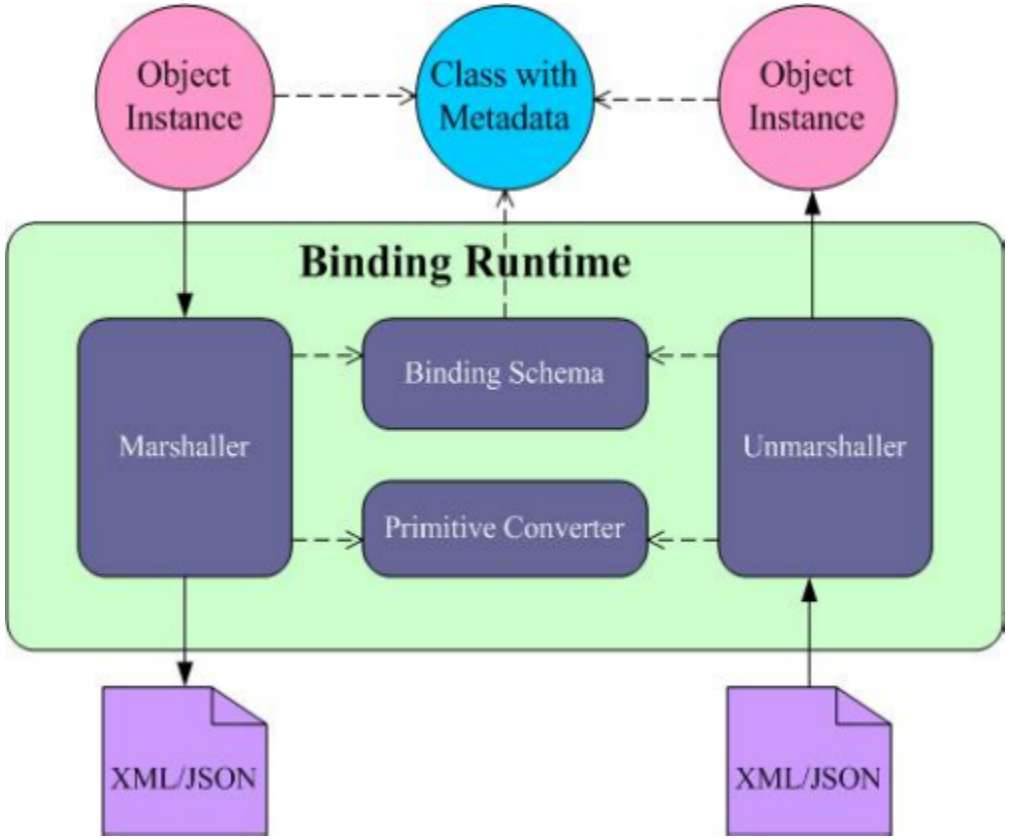
The Big Picture



Code Generation from WSDL



Automatic XML<>Object Binding



WSDL Driven Dev Flow on Android

1. Generate Android Java proxy from WSDL,
2. Create new Android project, add Nano 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. Callback object with success, failure and SOAP fault handling logic

```
/**
 Returns the non-zero dollar amount of the passed number.
*/
public void numberToDollars(NumberToDollars requestObject, SOAPServiceCallback<NumberToDollarsResponse> serviceCallback)
```

```
public interface SOAPServiceCallback<R> {

    public void onSuccess(R responseObject);

    public void onFailure(Throwable error, String errorMessage);

    public void onSOAPFault(Object soapFault);

}
```

A Service Call Sample

```
numberToWordsButton.setOnClickListener(new OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // get shared client
        NumberConversionSoapType_SOAPClient client = NumberConversionServiceClient.getSharedClient();
        client.setDebug(true); // enable soap message logging

        // build request
        NumberToWords request = new NumberToWords();
        String number = ((EditText)findViewById(R.id.numerInputText)).getText().toString();
        request.ubiNum = new BigInteger(number);

        // make API call and register callbacks
        client.numberToWords(request, new SOAPServiceCallback<NumberToWordsResponse>() {

            @Override
            public void onSuccess(NumberToWordsResponse responseObject) { // success
                Toast.makeText(MainActivity.this, responseObject.numberToWordsResult, Toast.LENGTH_LONG).show();
            }

            @Override
            public void onFailure(Throwable error, String errorMessage) { // http or parsing error
                Toast.makeText(MainActivity.this, errorMessage, Toast.LENGTH_LONG).show();
            }

            @Override
            public void onSOAPFault(Object soapFault) { // soap fault
                Fault fault = (Fault)soapFault;
                Toast.makeText(MainActivity.this, fault.faultstring, Toast.LENGTH_LONG).show();
            }

        });

    }

});
```

Demo 1 – BarCode

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:

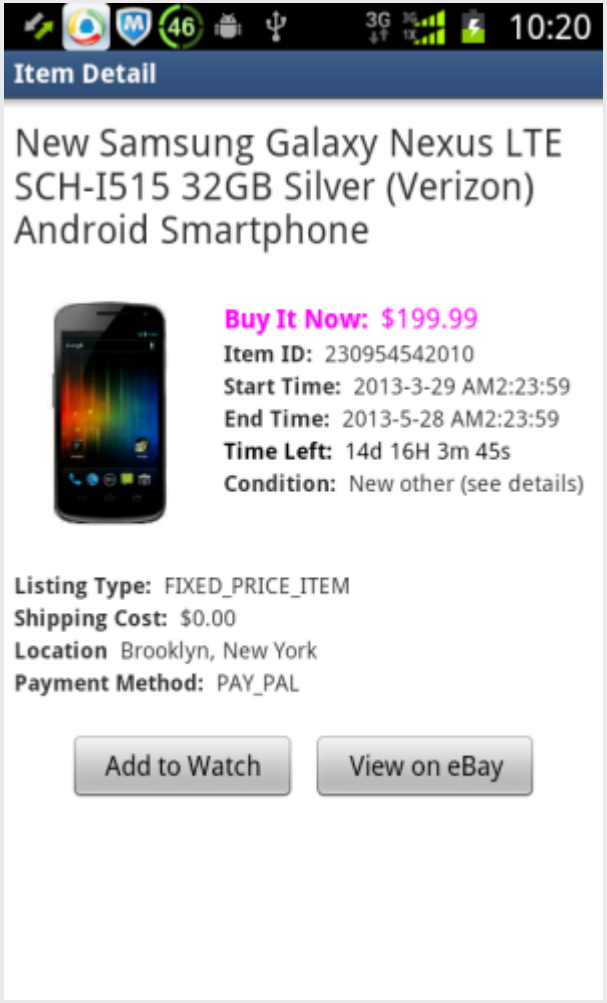
1. ***findItemsByKeywords*** for item search

eBay Shopping API used:

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

eBay Trading API used:

3. ***addToWatchList*** for adding item to watch list



Source, Samples and Tutorials

For Android:
https://github.com/bulldog2011/nano
Similar framework for iOS:
https://github.com/bulldog2011/pico