“Technology is nothing. What's important is that you have a faith in people, that they're basically good and smart, and if you give them tools, they'll do wonderful things with them.”

– **Steve Jobs**

## Introduction

The present document is focused on a best practice methodology for driving software processes, without expanding into vertical markets.

Formall technology architecture is based in a three tier client/server structure:

1. Presentation Layer
2. Business Layer
3. Persistence Layer

Here the business layer represents a single or multiple applications composing the business, which is agnostic of the persistence of its objects or their visual presentation, limiting it to its logic and inherent processes: separation of concerns.

All the applications share a common framework, the magic glue that connects the business objects with their persistence and presentation.

It is our meaning not to link this technology to specific implementation of persistence or presentation, which are quiet large in number of choices; for that we use a common driver model. Each driver will act as a wrapper around 3rd party implementation and expose a common set of features, or implement them if needed.

Our primary target is web applications. And the recent development of software technologies allows us to extend it to devices, at a low cost.

**Database**

**Microsoft SQL Server**

**Oracle Database**

**Oracle MySQL**

**Amazon DynamoDB**

**MongoDB**

**CouchDB**

**RavenDB**

**Browser**

**Twitter Bootstrap**

**Google Angular**

**Sencha Ext JS**

**Sencha Touch**

**Telerik KendoUI**

**JQuery UI**

**JQuery Mobile**

**Device**

**PhoneGap**

**Windows RT**

**Xamarin**

**Native**

**Framework**

**Driver**

**Driver**

**Business**

**Persistence**

**Presentation**

**Service**

**Twitter**

**Facebook**

**Google**

**Office360**

**Evernote**

**Dropbox**

# The Framework

"Some of the world's greatest feats were accomplished by people not smart enough to know they were impossible."

**– Doug Larson**

The framework to the application if what the dark matter is to all of us, and all around us. It connects every component, from the smallest and the largest. It gives meaning to existence, even if we are not aware of its own existence.

In the Object-oriented paradigm, concepts are represented as objects that have fields that describe the object, and associated methods.

A Business Object is a type of an intelligible entity being an actor inside the business layer. Business objects separate state from behavior because they are communicated across the tiers in a multi-tiered system, while the real work is done in the business tier and does not move across the tiers.

Technical Hint:

Here all business objects are POCO objects with no associated public methods. These objects may implement interfaces, but the properties and methods of those interfaces are implemented explicitly. In order to access such methods and interfaces the instance has to be cast to the interface itself.

The business object behaviors are implemented as extensions methods. Extension methods are a special kind of static method, but they are called as if they were instance of the method on the extended type. Their first parameter specifies which type the method operates on, and the parameter is preceded by the “this” modifier. This enables you to add methods to existing types without creating a new derived type.

## The Document

The document concept is defined as a preserved or recorded phenomenon, whether physical or mental, for reconstruction.

Since public business in the Middle Age was juridical in character, custumal or collections of customary law were the principal register of a medieval borough’s administration. And even before, documents have been used to register account state, inventory or transactions.

In or concrete definition, a document is the transitory state of an entity through the process of serialization (or deserialization.) A given document can double as “data access object” for persistence and “data transfer object” for presentation. It can also be used processes where the concrete type of the entity is not required.

We only define a set of interfaces located at Formall.Linq:

1. IDocument
2. IObject
3. IValidatable
4. IEntry
5. IDictionary
6. IValue
7. ICollection

We also define IDocumentContext inside the Formall.Persistence namespace, which will be explained later on the persistence section.

IObject interface

All frameworks define an object class as the top most ancestors of all classes. Here we define IObject for the same porpoise. Every document and smaller component implements this interface.

The main porpoise of IObject is serialization.

IDocument interface

The document is our largest unit. All the other interfaces defined on this namespace are in one way or another part of IDocument. IDocument only inherits from IObject, which is also the base interface for all the other interfaces on this namespace.

IValidatable interface

Further on the metadata section I will expose some classes will be used to validate each of these components/interfaces we are describing below and including IDocument.

IEntry interface

IDictionary interface

IValue interface

ICollection interface

## The Navigation

### Schema

Schema class

ISegment interface

Domain class

Area class

## The Metadata

“True simplicity is derived from so much more than just the absence of clutter and ornamentation. It’s about bringing order to complexity.”

**– Tim Cook**

Metadata refers to “data about data”. By describing the content of data, the quality of the original data is greatly increased. Structural metadata are used to describe the structure of a database object such as tables, columns, keys and indexes. Guide metadata are used to help humans find specific items and are usually expressed as a set of keywords in a natural language. Metadata can either represent technical metadata corresponding to internal metadata or business metadata corresponding to external metadata. Metadata can also describe processes.

Structural metadata give a description of how the components of the objects are organized. But it can also contain the name of the type it describes and information used to search and locate and object such as title, author, subjects, keywords, publisher, date of creation, last modified, etc.

The metacontent, or vocabularies used to assemble metadata statements, are structured according to a standardized concept using a well-defined metadata scheme.

The metacontent, or vocabularies used to assemble metadata statements, are structured according to a standardized concept using a well-defined metadata scheme.

Our metadata is located inside the Formall.Reflection namespace. We define a set of metacontent classes:

1. DataType
2. Value
3. List
4. Mode
5. Magnitude
6. Money

DataType class

Value class

List class

Item class

Model class

Field class

Action class

Magnitude class

Length

Mass

Unit class

Money class

An order of magnitude is a scale of numbers with a fixed ratio, often rounded to the nearest ten.

Currency class

## The Serialization

### JSON Format

### XML Format

## The Driver Model

“Give me a lever long enough and a fulcrum on which to place it and I shall move the world.”

**– Archimedes**

The driver is the muscle, the actuator. It transforms the meaning to action.

# The Business

"The great successful men of the world have used their imaginations, they think ahead and create their mental picture, and then go to work materializing that picture in all its details, filling in here, adding a little there, altering this a bit and that bit, but steadily building, steadily building."

**– Robert Collier**

## Business Object

"I am, I exist, whenever it is uttered from me, or conceived by the mind, necessarily is true."

**– René Descartes**

## Business Logic

"If you don't know where you are going, you'll end up someplace else."

**– Yogi Berra**

## Business Area

# The Persistence

“When you come to a fork in the road, take it!”

**– Yogi Berra**

## The Persistence Diver

“In theory, there is no difference between theory and practice. But in practice, there is.”

**– Yogi Berra**

IDocumentContext interface

## NON SQL Implementation

### File System

### Zip Package

### Raven DB

## SQL Implementation

### Microsoft SQL Server

### Oracle MySQL

# The Presentation

“Our imagination is the only limit to what we can hope to have in the future.”

**– Charles F. Kettering**

## The Presentation Driver

“I will not say I have failed 1000 times; I will say that I have discovered 1000 ways that can cause failure.”

**– Thomas Edison**

### Sencha Ext JS

### Sencha Touch

### Google Angular

### Twitter Bootstrap

## The Device Package

### PhoneGap

## The Syndication

## The View Template

“Logic will get you from A to B. Imagination will take you everywhere.”

**– Albert Einstein**

### Google Angular + Twitter Bootstrap

# Miscellaneous

## Using ASP.NET MVC

## Document and Message Exchange