

**C2C Web Service**  
Proposal  
(CRUD2CRUD Web Service Report)

Sr. #	Date	Subject	Version	Author
1	31-07-2021	CRUD2CRUD Web Service	1	Shams Zia

**Disclaimer:**

This document proposes a web standard named "C2C Web Service" and is a copyright material of Invincible Technologies, a products development company in Pakistan. Web Service and WSDL are W3C standards. Nuget packages and other intellectual material mentioned in this document is copyright of their respective companies or entities.

## Table of Contents

C2C Web Service.....	1
Abstract.....	3
Introduction.....	3
Architecture.....	3
Remote Invocations.....	3
Design.....	4
CRUD & CRUD2CRUD.....	4
CRUD2CRUD & Web Service.....	4
CRUD2CRUD Communication.....	5
Benefits.....	5
Adaptability.....	5
Extensibility.....	6
Comparison.....	6
Integration.....	6
Code Libraries.....	6
Examples & Tutorials.....	7
Conclusion.....	7

## Abstract

CRUD2CRUD simplifies web service sharing through unified interface as compared to Web Services Description Language (WSDL) where both client and services require interface, methods and data definitions to be advertised.

## Introduction

CRUD2CRUD is interface neutral and data centric, its data communication architecture. Client knows how to request and service knows what to response, there is no need of custom clients to be written against each service.

## Architecture

In CRUD2CRUD architecture, server side publish services using CRUD(L) controllers where as client side consume such services using pairing CRUD(L) controllers as explained in following diagram:

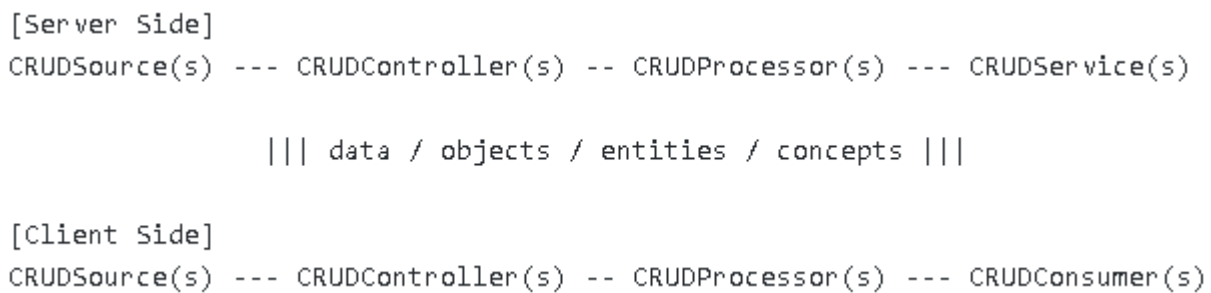


Fig. 1

Communication can take place at either level of design.

## Remote Invocations

CRUD2CRUD is source neutral where client does not know actual implementation, an invocation may last across multiple application boundaries forming a CRUD2CRUD2CRUD configuration.

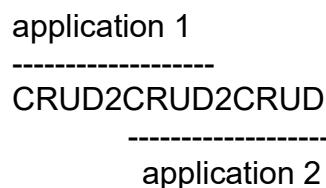
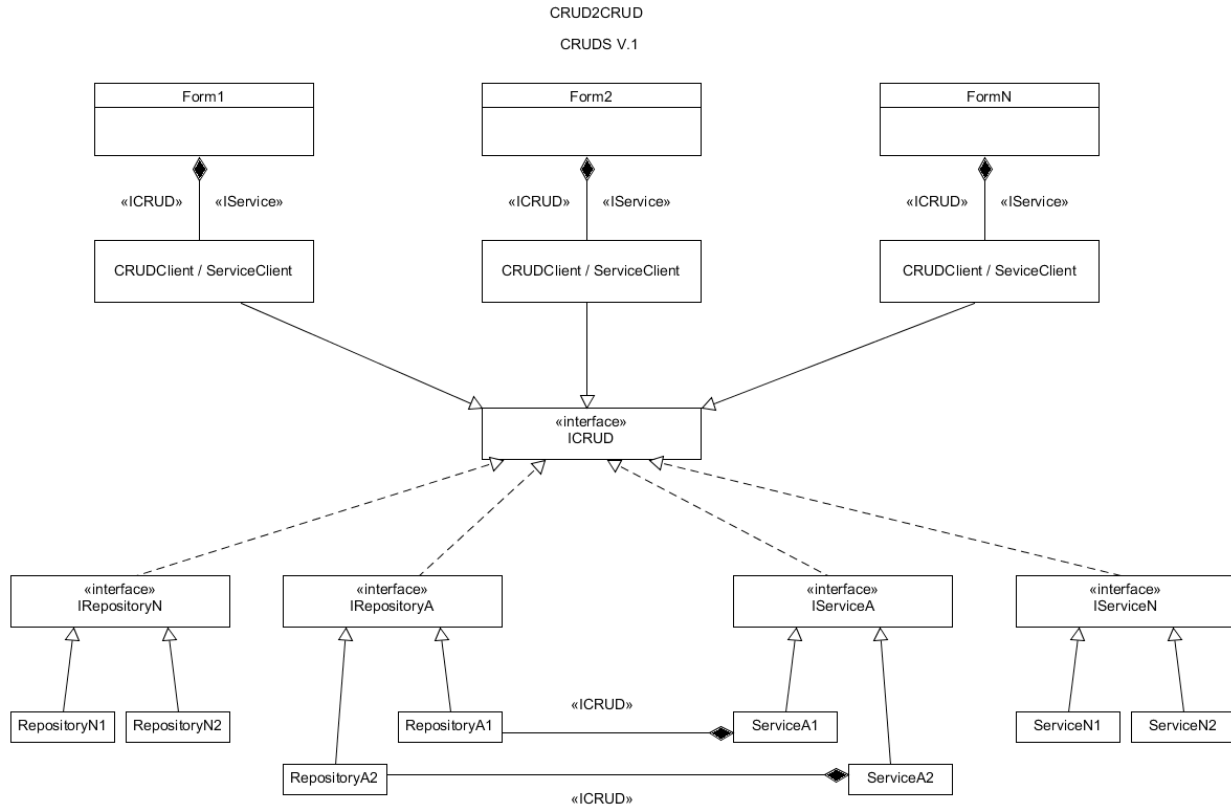


Fig. 2

## Design

CRUD2CRUD is an application of unified interface design.



CRUD is a data persistence reference design pattern where as CRUD2CRUD is a distributed application architecture that enables data communication on well defined CRUD interfaces across different subsystems of a distributed application.

Fig. 3

## CRUD & CRUD2CRUD

CRUD2CRUD is data centric and interface neutral where as CRUD services have concrete clients with specialized interfaces. CRUD is a data persistence reference design pattern where as CRUD2CRUD is an intent communication design pattern where persistence is not required.

CRUD2CRUD common interface for web and desktop lead toward pure data services. Programmers consume CRUD2CRUD data services without generating or implementing concrete clients.

## CRUD2CRUD & Web Service

In Web Services, WSDL explains SOAP client requirements, and specialized clients are implemented against each web service for program integrations, in case of CRUD2CRUD

communication architecture such specialized clients are not required.

Common interface lead programmers focus on data definitions, service setups and client integrations rather writing or generating individual API clients for each service.

For further reading about WSDL specification refer following link [Web Services Description Language \(WSDL\) Version 2.0 Part 0: Primer \(w3.org\)](#).

## **CRUD2CRUD Communication**

CRUD2CRUD is data architecture with intent communication. Developers focus on data either working with CRUD (CRUD2) client or a CRUD (2CRUD) service. CRUD2 client is data centric and source neutral, similarly 2CRUD service is data centric and client neutral.

In such an elaborated data architecture communication APIs become standard, and data required is published for integration by both clients and services. There is no need of writing specialized clients against each service and publish their APIs.

CRUD2CRUD with common interface enable engineers with communication data modeling that explains data models required for developing different sub-systems of a software application. Communication models are at the core of CRUD2CRUD applications. For example in a typical inventory application; Product, Sale, Purchase and Stock define data model of application, communication model(s) are subset of data model that vary according to sub-system, service or client implementation.

## **Benefits**

CRUD2CRUD (or CRUDS) bring following benefits.

1. Generalized Communication.
2. Understood Architecture with Common Interface.
3. CRUD Repositories for data persistence.
4. CRUDL communication with CRUD repositories.
5. Grows in repositories rather interface.
6. Invocation of extended (repository) methods using "target" attributes.
7. Reduced effort and cost of developing interfaces and method definitions.
8. Reduced complexity and elaborated architecture (CRUDL data communication).
9. Distribute applications with simple interfacing communication models.

## **Adaptability**

CRUD2CRUD is simple with understood architecture that attract programmers toward adaptability. Its highly productive, non-repetitive and requires less coding efforts.

Simple, futuristic with stable implementation, economical updates, programmer friendly, encourage design driven development, leading toward right visibility of progress, code

organization and management.

## Extensibility

CRUD2CRUD is extensible at no additional cost of development as it does not require interface methods and data type definitions.

## Comparison

Following compare CRUD2CRUD and WSDL (Web Service):

<b>CRUD2CRUD</b>	<b>WSDL (Web Service)</b>
Unified Interface	Custom Interface
Generalized Client	Specialized Clients, Each service requires a new client with specialized interface.
Encourage growth in repositories.	Grows in interface.
Shared Communication Model	Requires data definitions.
CRUDM (CRUDMethod) scales definition requirements.	Requires method definitions.
Understood common functionality (Create, Read, Update, Delete, List).	
Requires no regeneration.	Requires regenerating of clients on each extension.

## Integration

In CRUD2CRUD clients does not require customized code generation where as in Web Services, WSDL explains client side interface and methods.

## Code Libraries

CRUD2CRUD have following implementations:

<https://www.nuget.org/packages/WindnTrees.Abstraction/>  
<https://www.nuget.org/packages/WindnTrees.Core/>  
<https://www.nuget.org/packages/WindnTrees.ICRUDS/>  
<https://www.nuget.org/packages/WindnTrees.ICRUDS.Standard/>  
<https://www.nuget.org/packages/WindnTrees.CRUDS/>

## Examples & Tutorials

CRUD2CRUD have following tutorials and coding examples:

Web:

<http://www.invincibletec.com/Tutorial/Index/windntrees-web>

<http://www.invincibletec.com/uploads/references/zip/crudm-request-target-and-repository.zip>

Windows:

<http://www.invincibletec.com/Tutorial/Index/windntrees-windows>

<http://www.invincibletec.com/tutorial/detail/windnTrees-crudview-and-wpf-forms>

<http://www.invincibletec.com/uploads/references/zip/crudview-and-wpf-forms.zip>

Mobile:

<http://www.invincibletec.com/Tutorial/Index/windntrees-mobile>

<http://www.invincibletec.com/uploads/references/zip/xamarin-forms-and-crud-method.zip>

## Conclusion

CRUD2CRUD is interface neutral, data centric and communication architecture. W3C must devise a working group to establish C2C Web Service standard on similar basis of SOAP.