Common Component Specification Proposal

Date	2015-08-26
Spec Title	Cloud Data Object (CDO) (CCS-CDO-API)
Author	Peter Judge, Progress Software
Status	Supersedes v1.0 at https://github.com/CloudDataObject/CDO
Version	1.1

Executive Summary

This document contains a proposal to the Common Component Standards (CCS) Steering Committee to form a new project team whose goal it is to produce a specification for a **Cloud Data Object (CDO)** to define a data source component that allows access to service resources and business logic via

- Schema exposed by the service through its various resources,
- Individual read and write operations on the underlying resources; these are Create, Read, Update, Delete, collectively known as CRUD,
- Batch update operations (Submit) which may contain multiple records that have been updated and/or deleted and/or added,
- Other developer-defined publicly-exposed operations which are business logic-dependent (collectively known as Invoke operations) and operate on a resource.

The CDO provides an API that can be used to work with the data locally and synchronize with the service when needed.

The CDO may optionally consume a *CDS Catalog (CCS-CDO-CAT)* which documents a service's the resources and logic in a similar form.

Component Description

A Cloud Data Service (CDS) defines the API for one or more server-side resources. Each resource provides access to a logical schema and its related business logic, through a set of operations, which may be a standard (CRUDS) or developer-defined (Invoke). These operations and schema may be described using a *CDS Catalog* (CCS-CDS-CAT).

The *CDO* is a component that provides access to a single resource (its schema & operations) from a client. A client here should be understood to mean "something

that consumes a server resource" and no more (ie no UI or other assumption); this assumption allows the CDO to fit into multiple layers of an application architecture, including UI on the client- or server-side, or even as a middleware or messaging component.

Historically, the CDO has been implemented in JavaScript (cf. https://github.com/CloudDataObject/JSDO).

Benefits and Use Cases

The *Cloud Data Object* is language agnostic and is useful when the client and the server are implemented in different technologies. This makes *CDO* extremely useful for integrating new client apps into existing business solutions.

The *Cloud Data Object*

- Supports local standard (CRUD) operations on a resource's schema,
- Supports arbitrary / developer-defined operations,
- Provides transactional support for those operations (ie rollback), both in the client as well as the capability to provide that data to service resources,
- Supports automatic and manual synchronization of resource, allowing support of offline environments,
- Supports multi-level resource schemas, (ie matching a relational schema) including nested relations and their cardinatilty,
- Maintains a session and related context (login/security, etc), including knowledge of any uncommitted changes,
- Supports the capability for complex retrieval operations, including but not limited to filtering, sorting, paging/batching and
- Is transport protocol agnostic, meaning no constraints on the implementation technology.

The *Cloud Data Object* may also

- Consume a *Cloud Data Service Catalog (CCS-CDS-CAT)* as the basis for its operations and schema,
- Act as a data-source component for a browser-based JavaScript UI framework, for example KendoUI or Angular.js

The *Cloud Data Object* is not

- Tied directly into any UI (via a browser DOM or similar mechanism)

Related / Dependent Common Component Specifications

The CDO MAY use a *CDS Catalog (CCS-CDS-CAT)* to automatically provide (import) schema and supported operations.

The CDO MAY use the *CDO Protocol (CCS-CDO-PRO)* for transporting messages to and from a service/resource.

Project Team Requirements

A CDO team has not yet been formed. The team's formation will commence upon acceptance of this standards specification proposal by the CCS Steering Committee. The goal of this project team will be to submit a Community Review Draft of the Specification to the CCS Steering Committee within 90 days after the team is formed. If the specification is accepted, it will be published to the entire CCS participant list for review.

Deliverables

- A formal specification of the CDO API (programming interface)
- Samples of an implementation of the CDO API, in JavaScript. This will most likely take the form of updates (via a fork and/or branch) to the existing JSDO implementation available publicly at https://github.com/CloudDataObject/JSDO .