

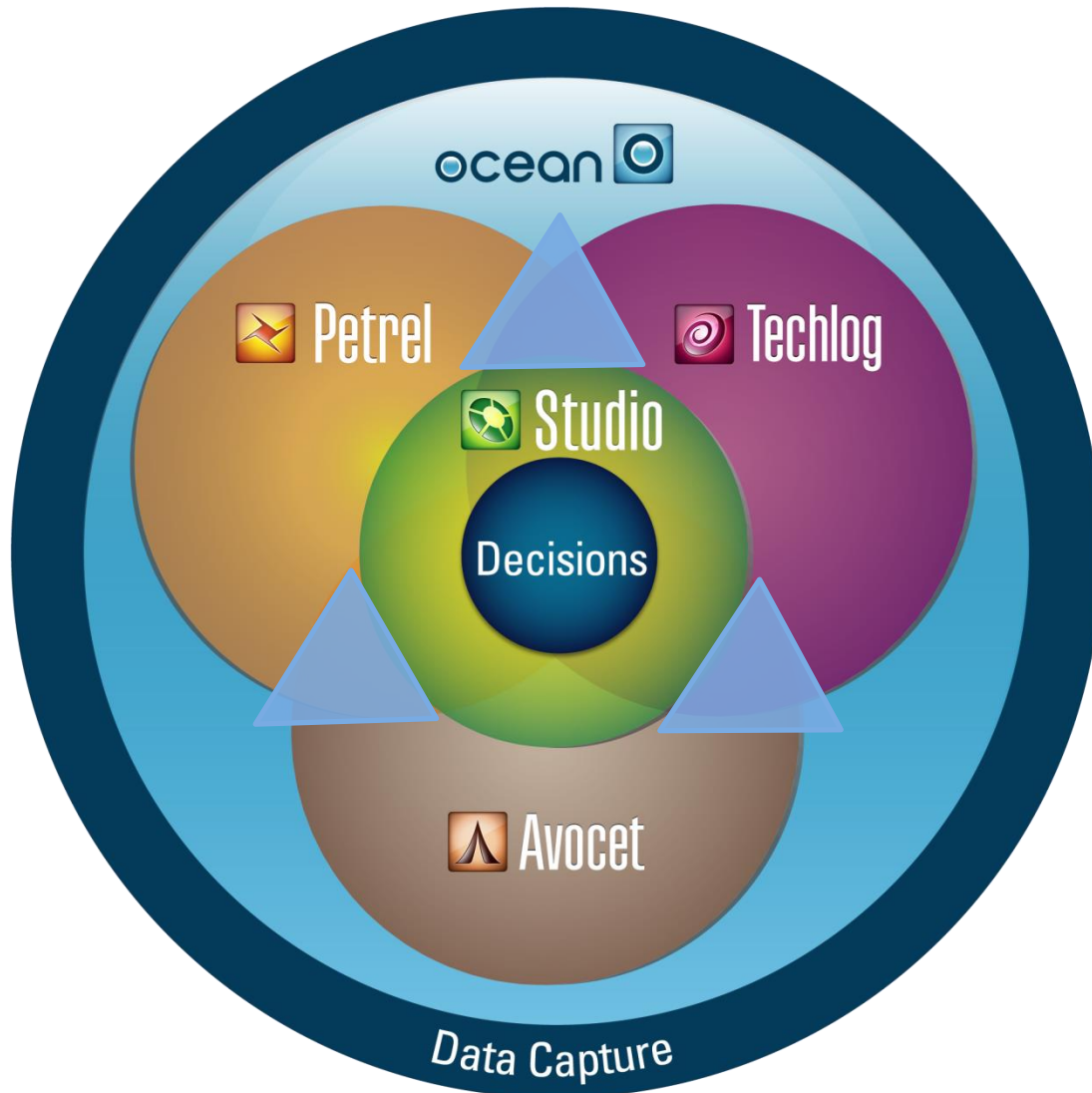
Ocean Framework for Studio Extending the Ecosystem

Gary Murphy

Ocean for Studio Product Champion

Ocean UGM 2014, Amsterdam

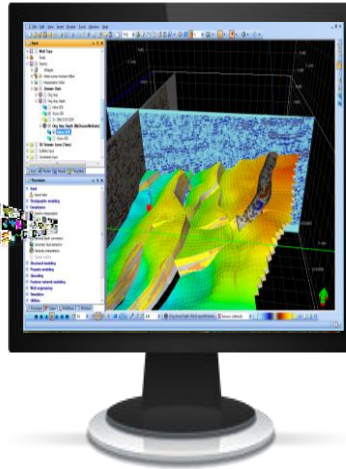
Schlumberger



What Is Ocean?



ocean 
Creating advantage



Platform

*In-House software, simulators, New seismic attributes
Modeling algorithms, Optimizers Visualizers...
New disciplines, new workflows, New methods
Hundreds of connectors*



Schlumberger

Ocean for Studio Goals

Extend Ocean Concept to Studio in order to:

- Deliver Platform Extensibility and Openness to Clients
- Ensure New Studio Capabilities Can Be Developed Quickly
- Enable Innovative Workflows and New Approaches
- Locate Data-Centric Workflows at the Right Place

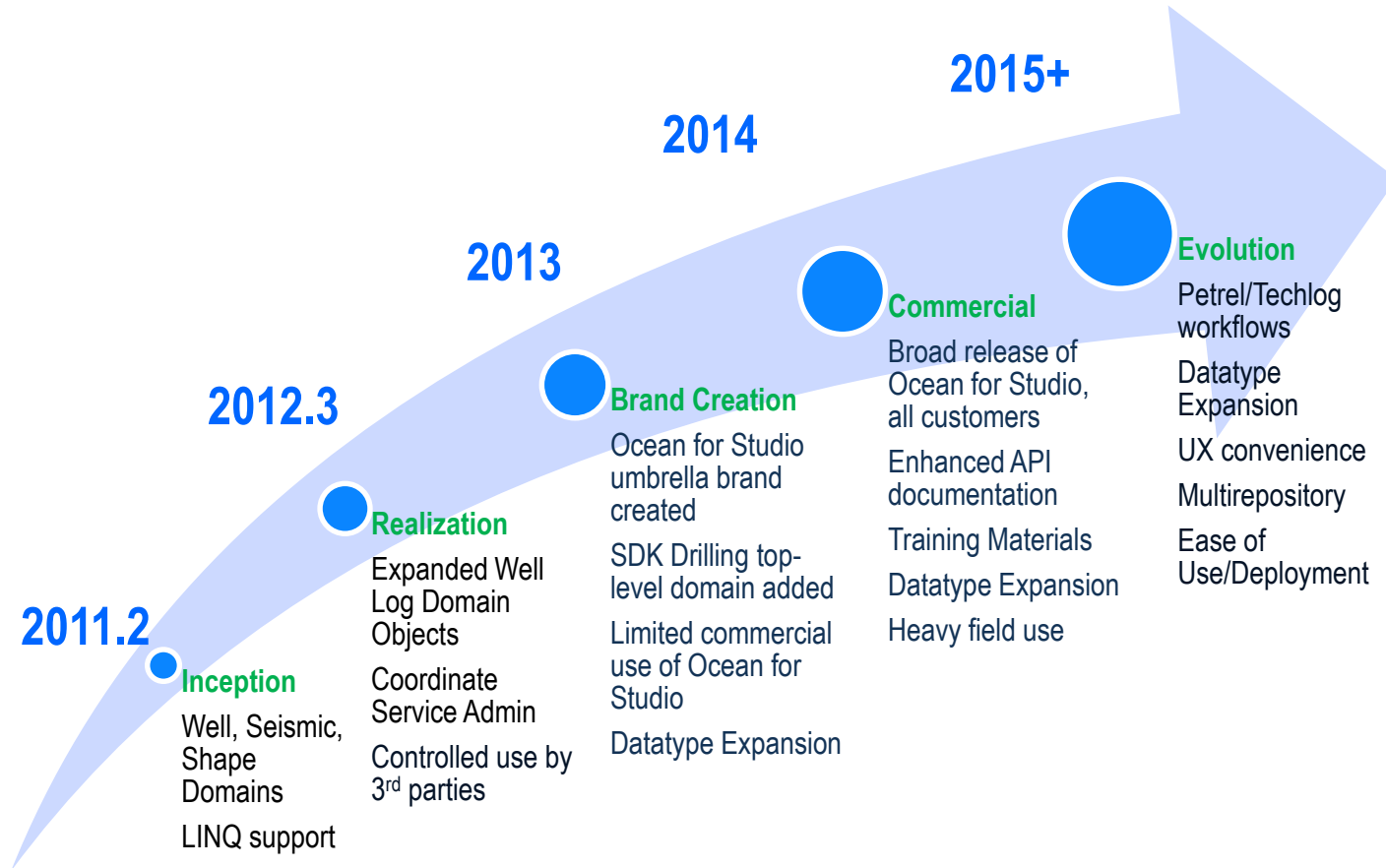
Ocean for Studio: Three API Families



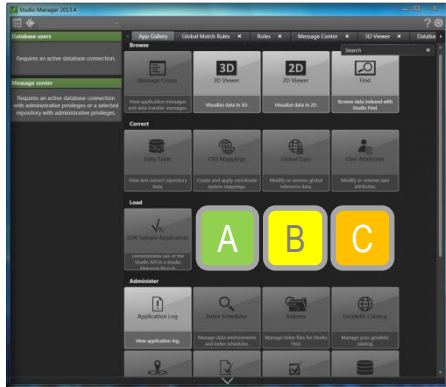
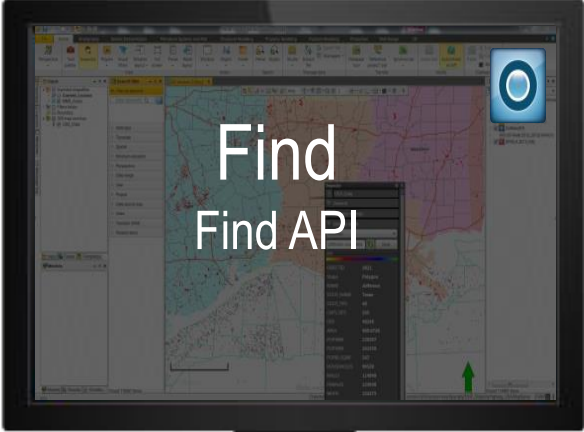
Ocean for Studio

- Find API
- Database SDK
- Studio Manager API

Ocean for Studio Evolution



Ocean for Studio Components



Find API



Consume



Index 1

Index 2

Generate

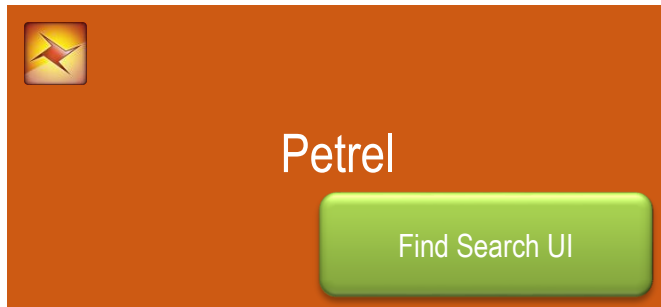
Plug-in 1

Plug-in 2

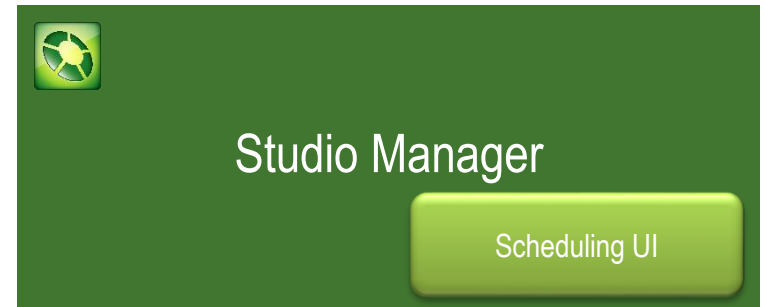
Studio Manager



Studio Find Overview

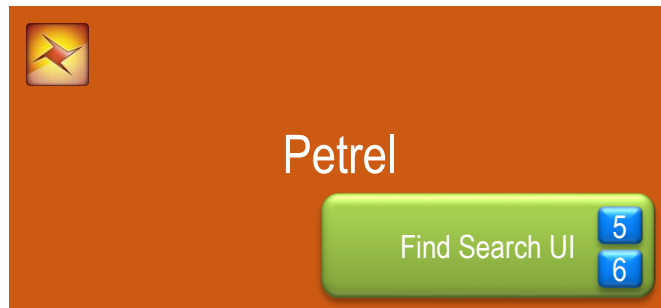


Search/End user workflow

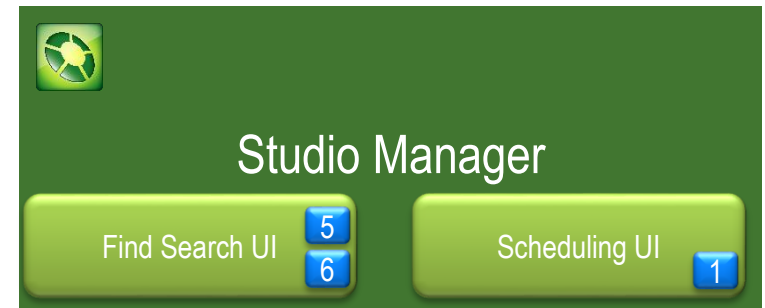


Index generation/Administration

Studio Find Extension Points



Search/End user workflow



Starts

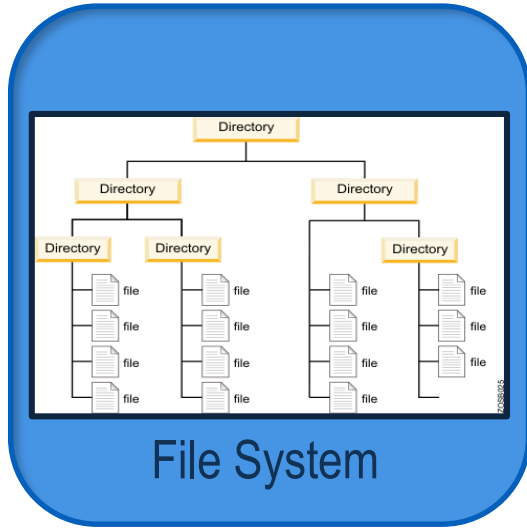
A red arrow points from the Scheduling Windows Service down to the Indexer.



Index generation/Administration

- 1 GUI for defining ABC datasource connection
- 2 (optional) Change monitor detector which detects when ABC data has changed
- 3 Actual indexer which reads data from ABC and publishes it to Find index
- 4 (optional) Augmenter which can enrich data indexed by other data vendors
- 5 (optional) Loader which can read data from ABC and load it into Petrel
- 6 (optional) Commands which user can activate to drill down into related data etc

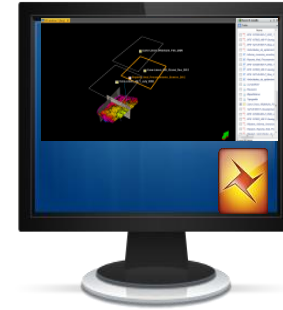
Example: Seismic Documents



File System



Database



In Context Data



Find Index Demo

Database SDK



Studio Manager



Plug-in



Database SDK



Other



Studio Runtime

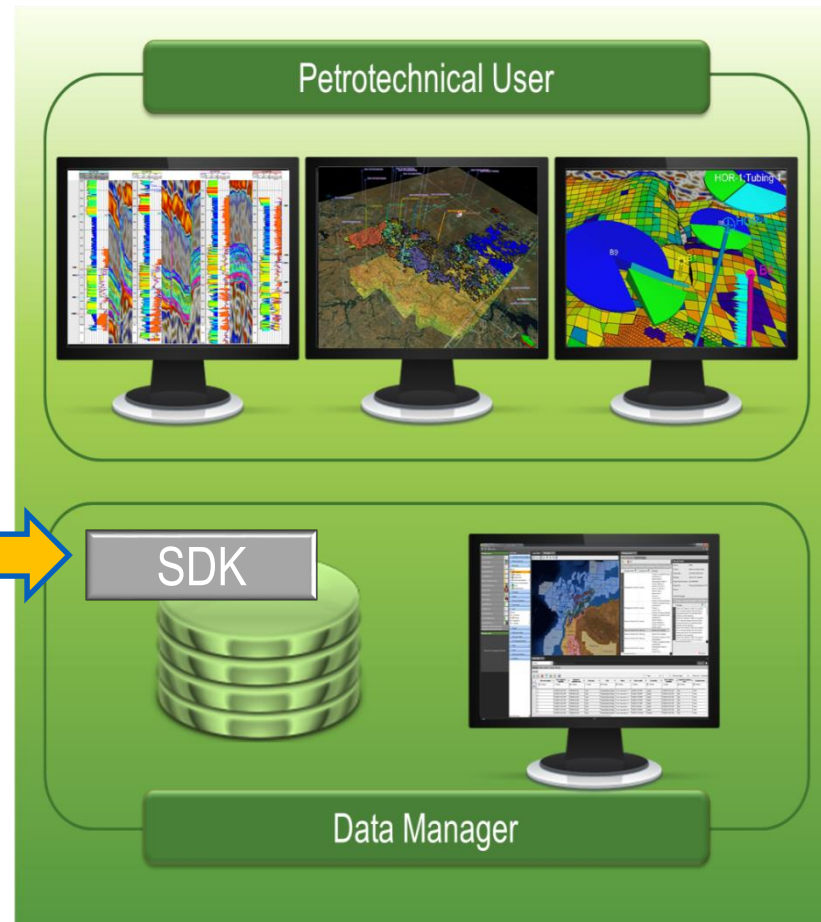


Database SDK



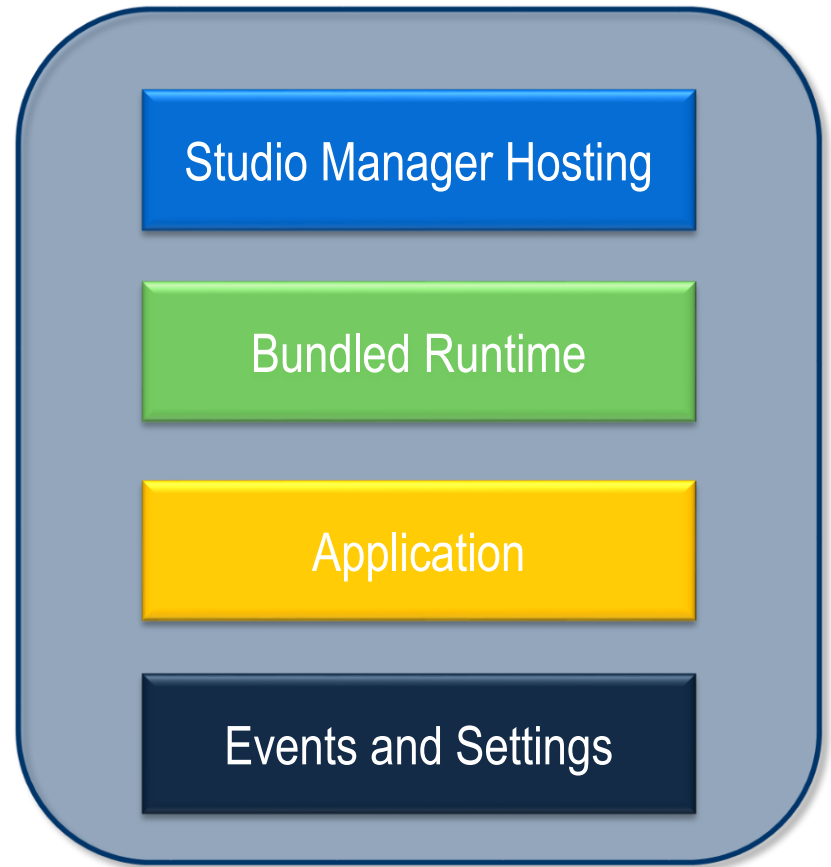
Studio

Studio SDK



Studio SDK Demo

Studio Manager API



Studio Manager API – Plug-ins

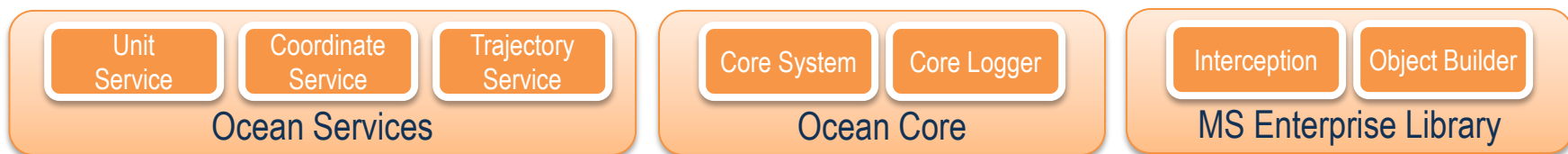
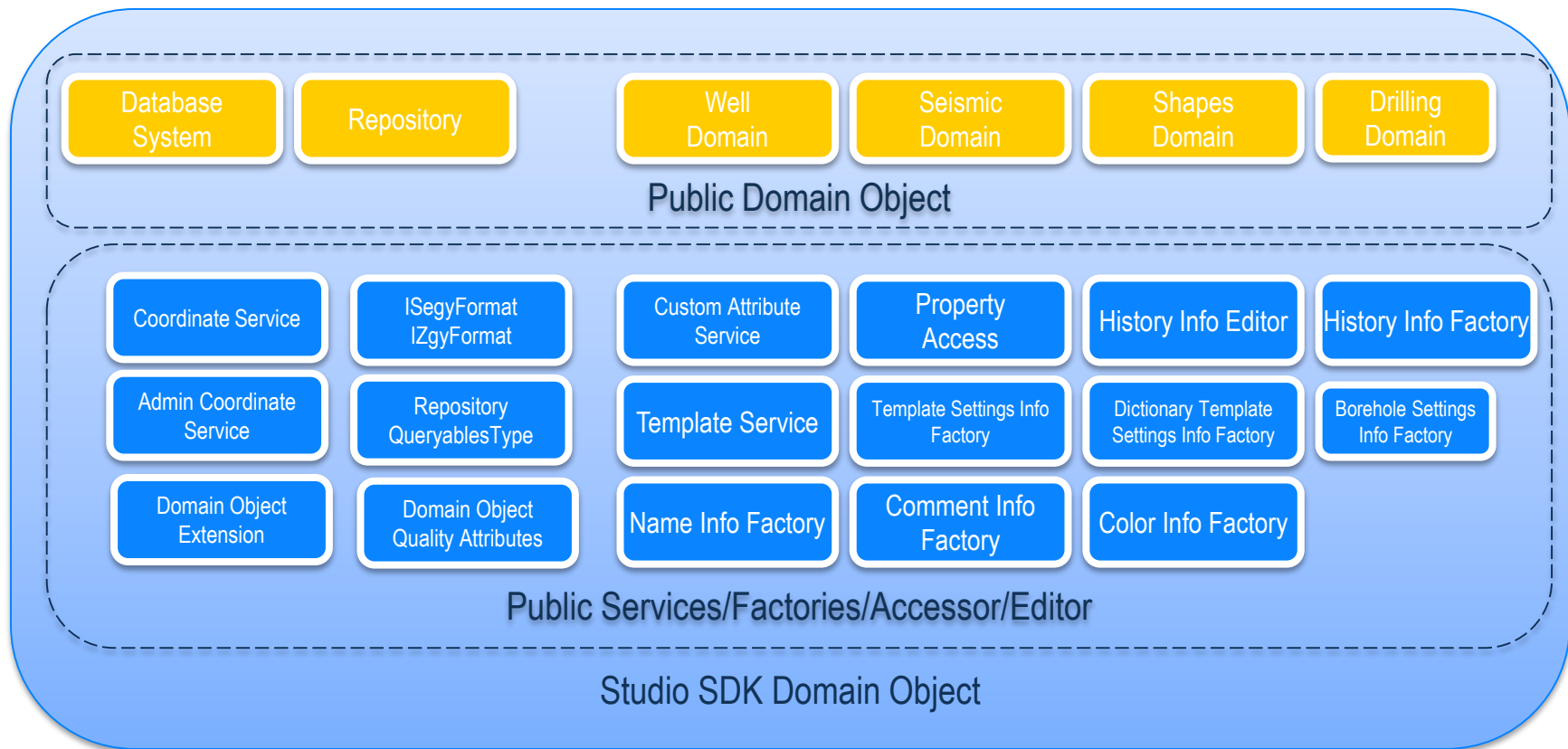
The screenshot displays the Studio Manager 2013.4 interface. The window title is "Studio Manager 2013.4". The interface is divided into a left sidebar and a main content area. The sidebar contains two sections: "Database users" (with the note "Requires an active database connection.") and "Message center" (with the note "Requires an active database connection with administrative privileges or a selected repository with administrative privileges."). The main content area features a top navigation bar with tabs for "App Gallery", "Global Match Rules", "Roles", "Message Center", "3D Viewer", and "Database". Below the navigation bar, the interface is organized into several functional groups:

- Browse:** Includes "Message Center" (View application messages and data transfer messages), "3D Viewer" (Visualize data in 3D), "2D Viewer" (Visualize data in 2D), and "Find" (Browse data indexed with Studio Find).
- Correct:** Includes "Data Table" (View and correct repository data), "CRS Mappings" (Create and apply coordinate system mappings), "Global Data" (Modify or remove global reference data), and "User Attributes" (Modify or remove user attributes).
- Load:** Includes "SDK Sample Application" (Demonstrates use of the Studio API in a Studio Manager Plug-In) and three colored buttons labeled "A1" (green), "A2" (yellow), and "A3" (orange).
- Administer:** Includes "Application Log" (View application log), "Index Schedules" (Manage data environments and index schedules), "Indexes" (Manage index files for Studio Find), and "Geodetic Catalog" (Manage your geodetic catalog).

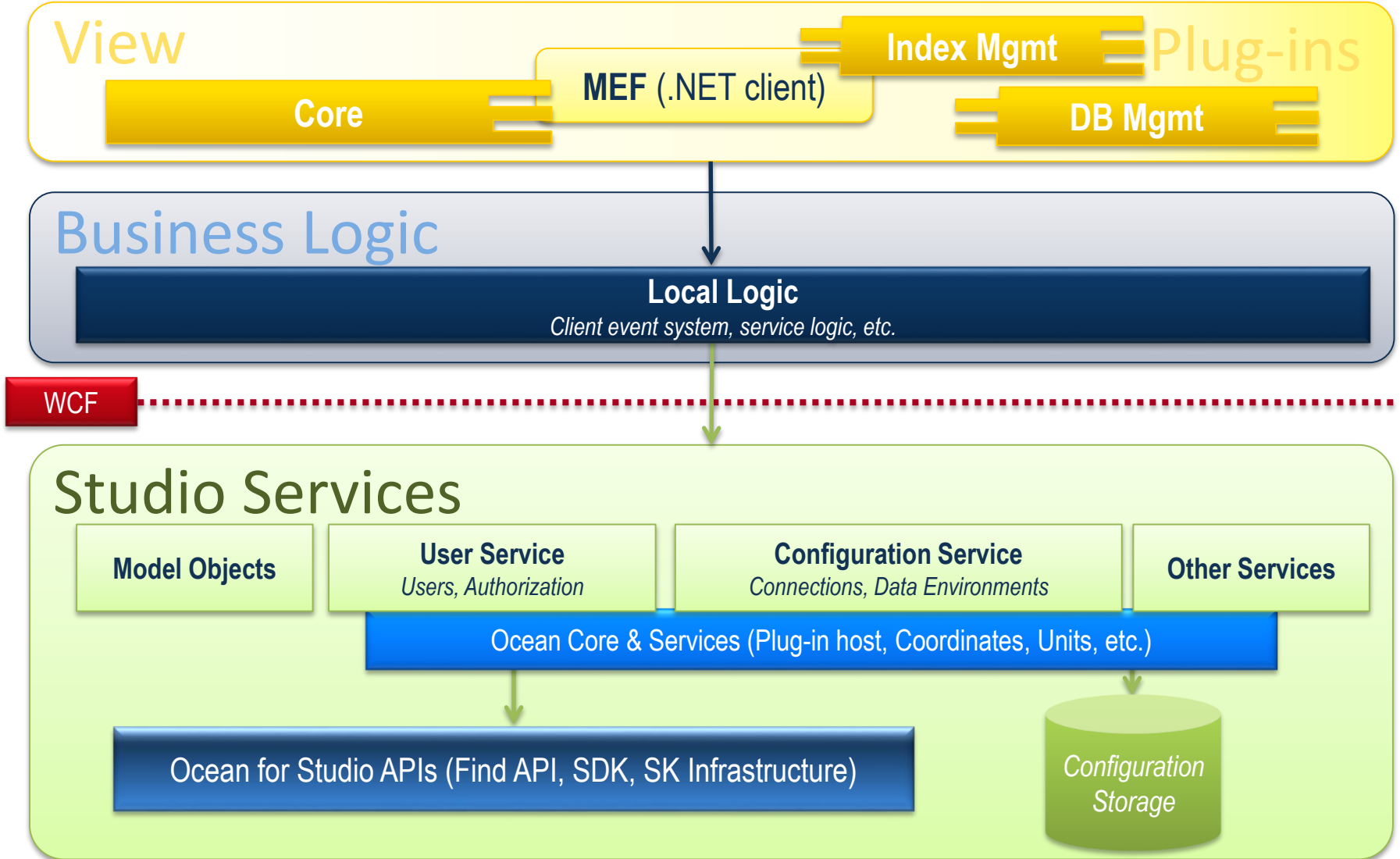
At the bottom of the interface, there are four additional icons: a location pin, a document with a checkmark, a checkmark, and a database cylinder.

Studio Manager Demo

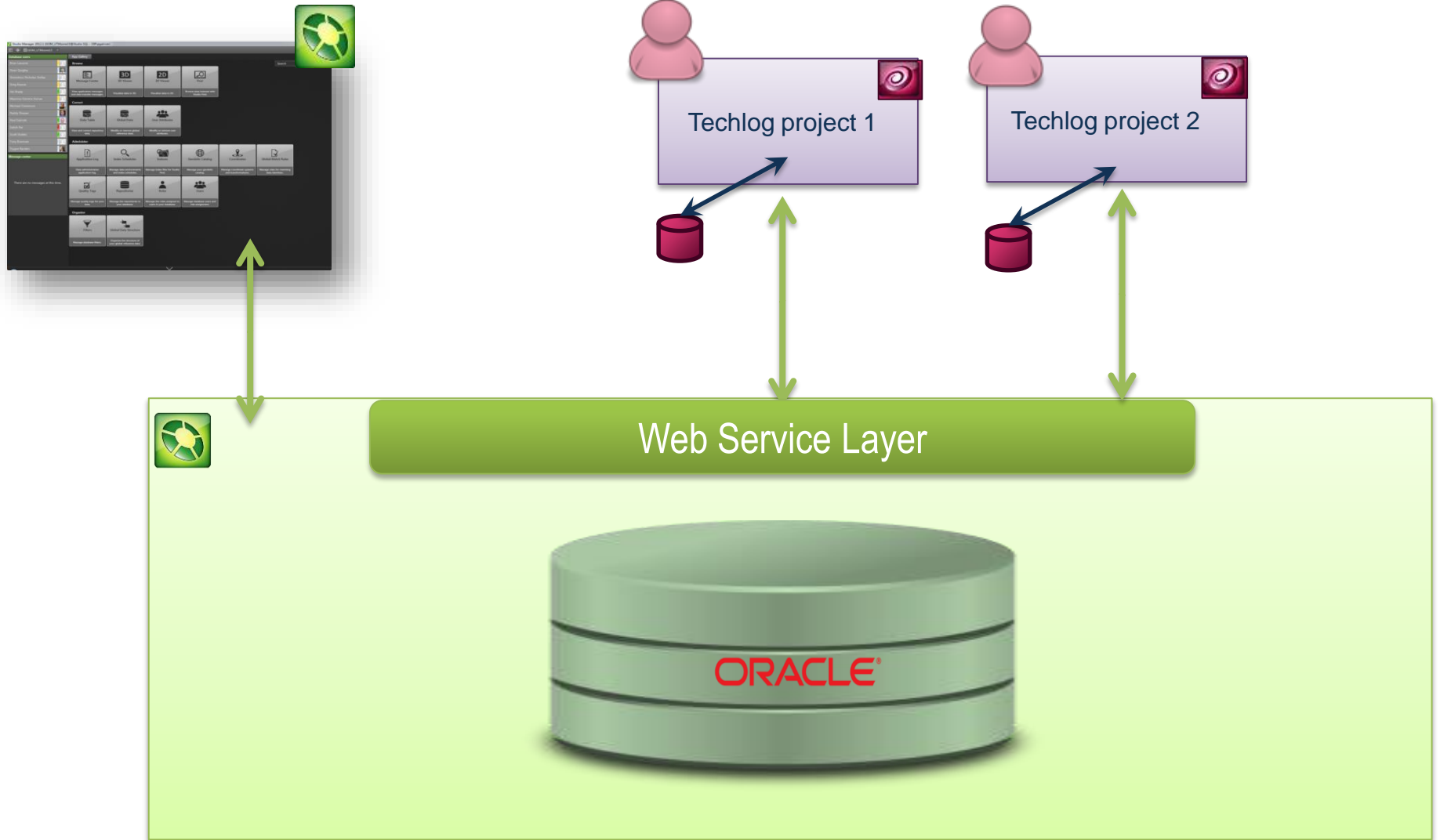
Architectures



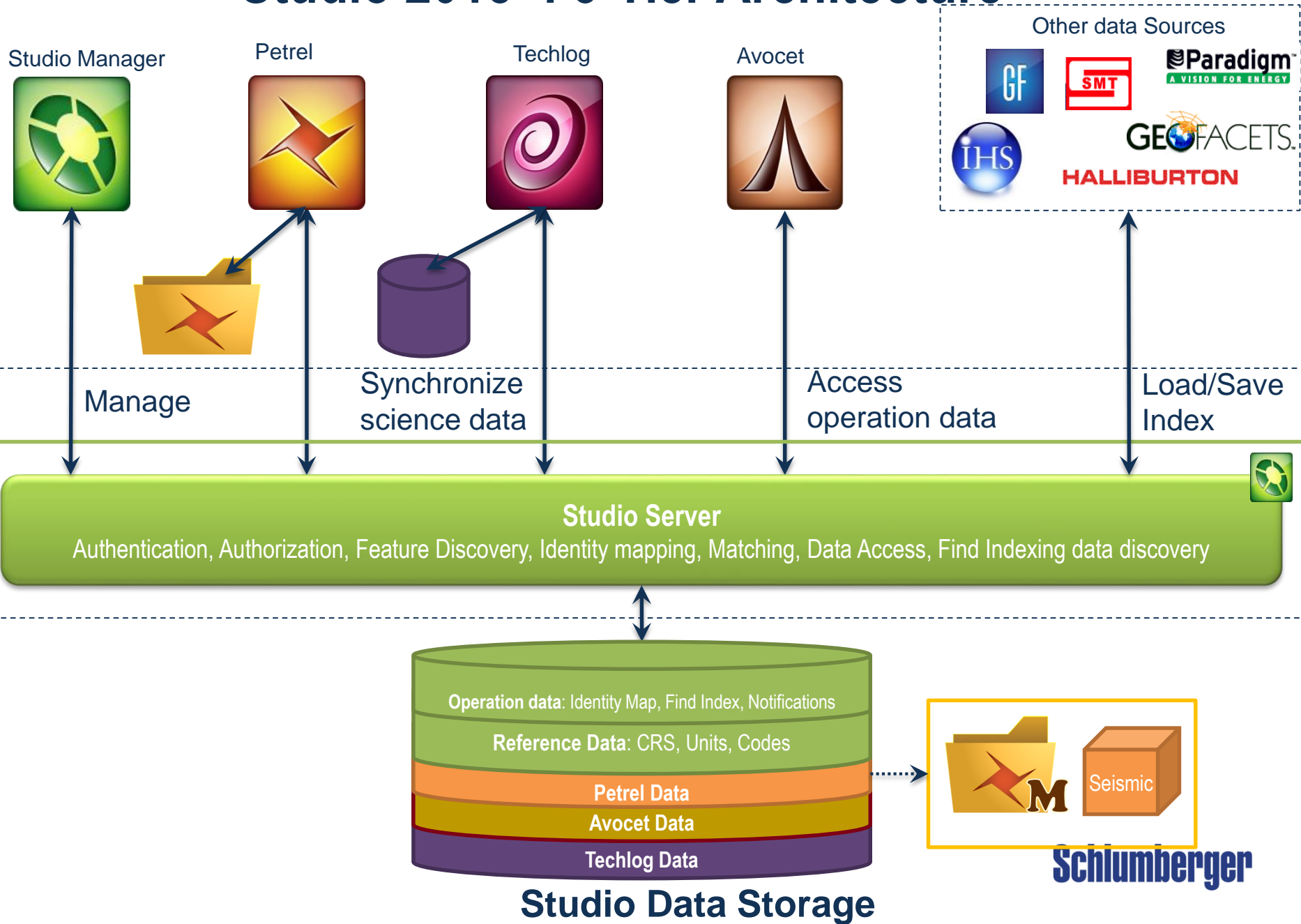
Studio Manager Architecture – 2013+



Techlog Architecture 2014



Studio 2015+: 3-Tier Architecture



Studio Architecture Directions

- Server-side plug-in architecture
 - Rich server-side infrastructure (server-side Ocean components):
 - Logging, tracing, performance monitoring
 - Authentication, authorization
 - Identity matching/mapping
 - Data access, Data validation
 - Indexing, searching, data analysis
 - Frame of Reference conversion
 - Publish/subscribe infrastructure
 - Platform extensibility scales down to single desktop, up to large server farm
- Client applications (Petrel, Techlog) integrate Studio functionalities natively
 - Support for centralized management and distribution of Petrel-Ocean plugins
 - Studio Manager becomes web first (HTML 5)
 - Allows support for multiple devices
 - Separation of business logic from GUI in plugins

2014 Deployment

Schlumberger

Pricing

- All Ocean Platforms for One Price – **One Ocean** Discount for Ocean Store plug-in Deployment

Licensing

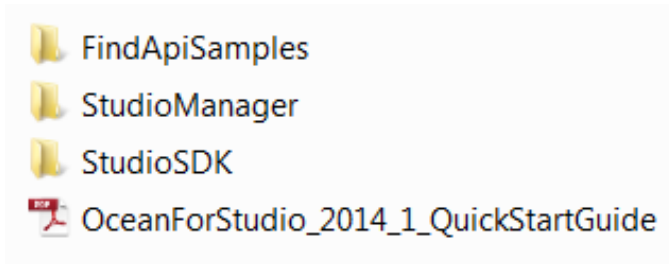
- Studio Runtime included with Studio Manager
- Separate Runtime License for non-SM Applications

Availability

- Commercially Available in Studio 2014.1

Physical Packaging and Licenses

- Packages delivered with the Studio DVD
 - Not via an installer, from a zip file = OceanForStudio_2014.zip
- Unzipped you get a set of directories



- License can be either Commercial or Development
- Development creates limited, or “tainted” repositories

Documentation and Samples





- Full set of CHM API files, one for each Ocean for Studio sub-system
- SDK Developer's Guide
- Studio Manager Developer's Guide
- Style Guide for Studio Manager Plug-ins
- Sample Programs
 - Find Indexers
 - SDK standalone and Studio Manager Loader

[-] **Namespaces**

Namespace	Description
Slb.Ocean.Studio.Find	Contains top level classes, e.g. IndexingContext .
Slb.Ocean.Studio.Find.Index	Contains the classes and services to support indexing of different data sources in Studio environment. <ul style="list-style-type: none">• See IndexerSystem for how to use the classes in this namespace when implementing a Indexer to crawl data and add it to the index.• See the Slb.Ocean.Studio.Find.Index.Administration namespace for how to implement a plug-in to the index administration application which will let the administrator specify which data sources should be indexed.• See the DaemonSystem namespace for how to implement a plug-in into the Daemon process so that a plug-in can notify Find when its data source has changed.
Slb.Ocean.Studio.Find.Index.Administration	Contains the classes and services to manage indexing configuration of different data sources in Studio environment. And provides abstract classes that allow developer to implement a plug-in to the index administration application which will let the administrator specify how data sources should be indexed.

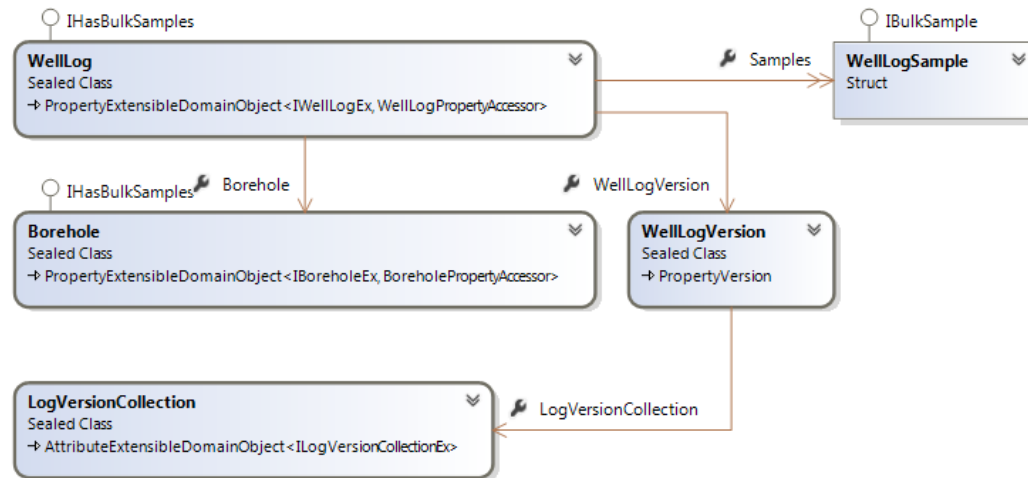
WellLog Class

[Ocean Studio Knowledge Data](#) ▶ [Slb.Ocean.Studio.Knowledge.Data.Well](#) ▶ **WellLog**

	Samples	Sets and gets sample elements (WellLogSample) through an enumerator. Note that the set method updates the entire WellLog . The new size will correspond to the size of the array, on which the enumerator iterates.
	UniqueId	Gets the unique id. (Inherited from DomainObjectBase .)
	Validate(List<ValidationResult>)	Validates the specified results. (Inherited from DomainObjectBase .)
	WellLogVersion	Gets the WellLogVersion for this WellLog.

Remarks

This diagram shows the specific support for Well Logs available in the Studio Knowledge SDK



Examples















Create WellLogs

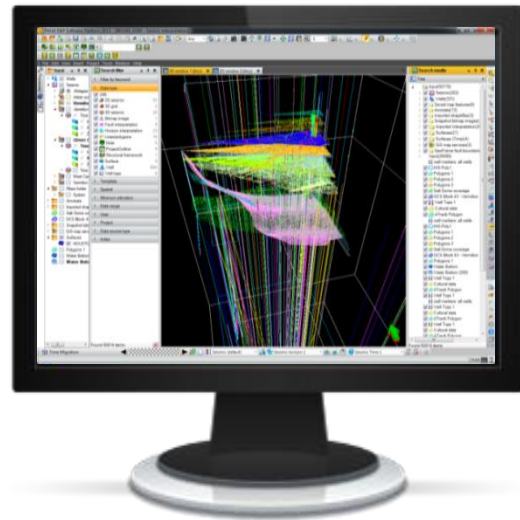
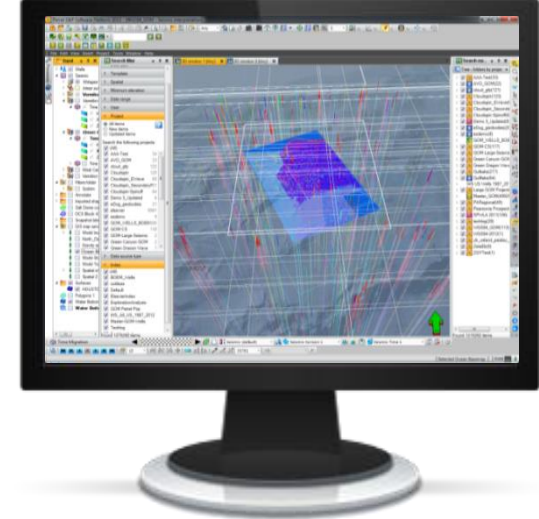
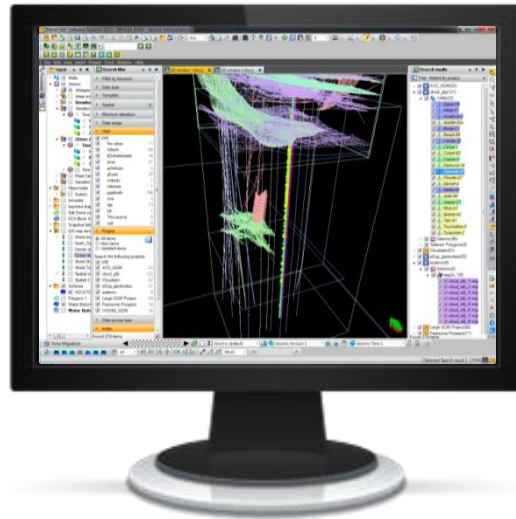
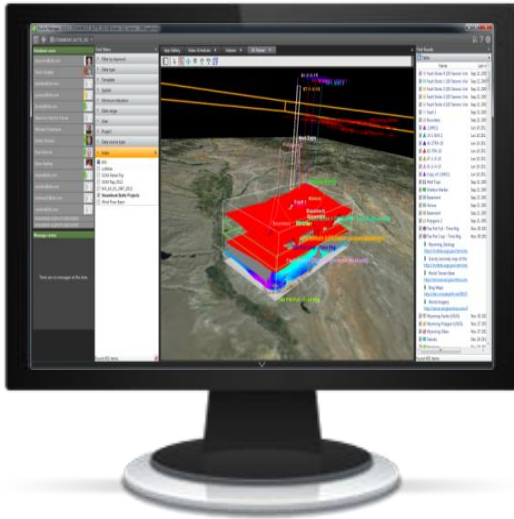
Slb.Studio.Manager.Apps Namespace

Studio Manager 2014.1 AF

[SubjectToChange] Contains classes that provide methods and properties to create Studio Manager Apps.

Classes

Class	Description
 ActiveAppChangedEvent	[SubjectToChange] Represents the ActiveAppChanged event.
 ActiveAppChangedEventPayload	[SubjectToChange] Event arguments for the ActiveAppChanged event.
 App	[SubjectToChange] Provides the base class implementation for an App in Studio Manager. Should be registered by Plugins wishing to extend Studio Manager app UI.
 AppCatalogFilter	[SubjectToChange] A filter for the app catalog.
 AppCategory	[SubjectToChange] A top-level application category in Studio Manager.
 AppInfoAttribute	[SubjectToChange] Specifies information about an App that can be used before loading the app.
 AppToolBarChangedEvent	[SubjectToChange] Triggered when an app's toolbar changes.
 AppToolBarChangedEventPayload	[SubjectToChange] Payload for AppToolbarChangedEvent.
 AvailableAppsChangedEvent	[SubjectToChange] Event triggered when the available apps in Studio Manager change.
 AvailableAppsChangedEventPayload	[SubjectToChange] Payload for AvailableAppsChanged event.
 RunningAppsChangedEvent	[SubjectToChange] Event triggered when the running apps in Studio Manager change.
 RunningAppsChangedEventPayload	[SubjectToChange] Payload for RunningAppsChanged event.
 SimpleApp	[SubjectToChange] A simple abstract app. Inherit to expose a new app to Studio Manager.
 WellKnownAppCategories	[SubjectToChange] Predefined application categories in Studio Manager.



Ocean for Studio is the Path to Data Advantage

