



Infinite Integration

OpenSpirit Next Generation Dev Kit

(available in v3.1.1+)

July 2008 Webcast

Clay Harter

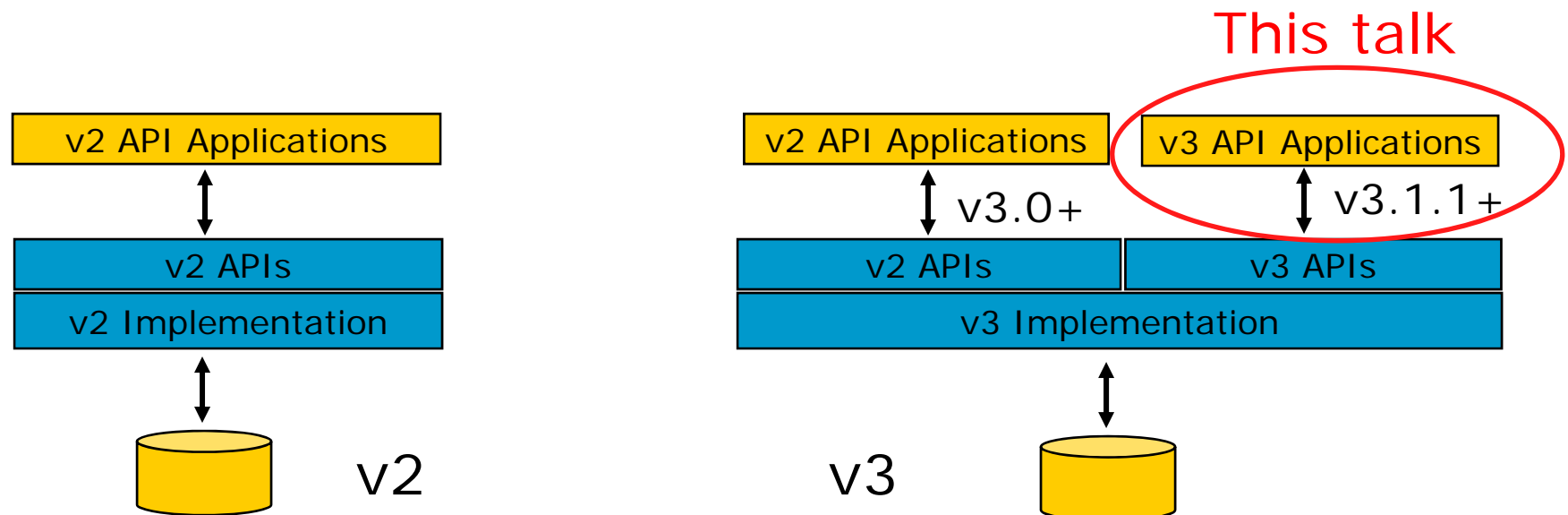


v2 to v3 Migration

V3 re-implements all key OpenSpirit services and provides many new capabilities and benefits.

V3.0 released Dec 2006

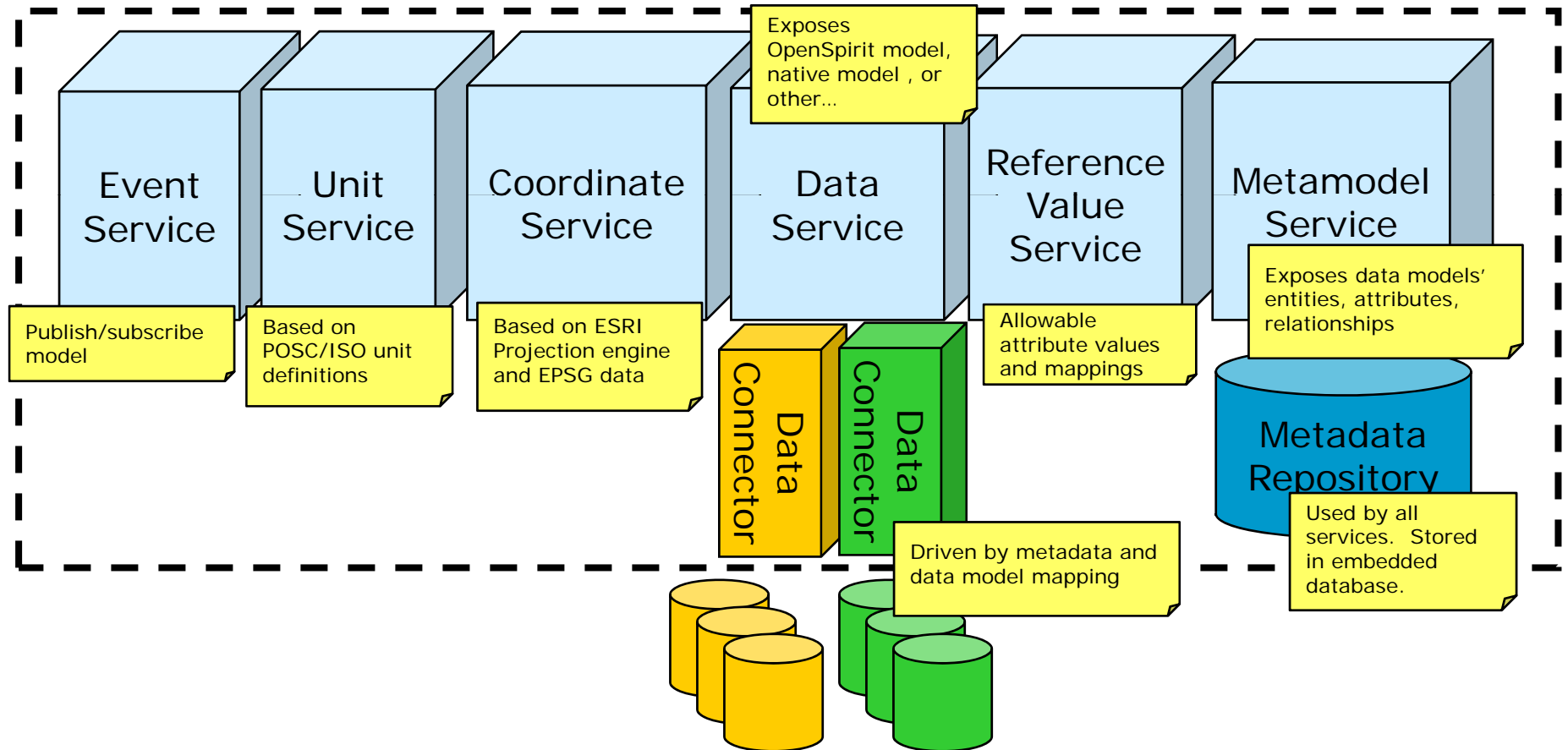
V3.1.1 released March 2008 (3.1.2 is current release)



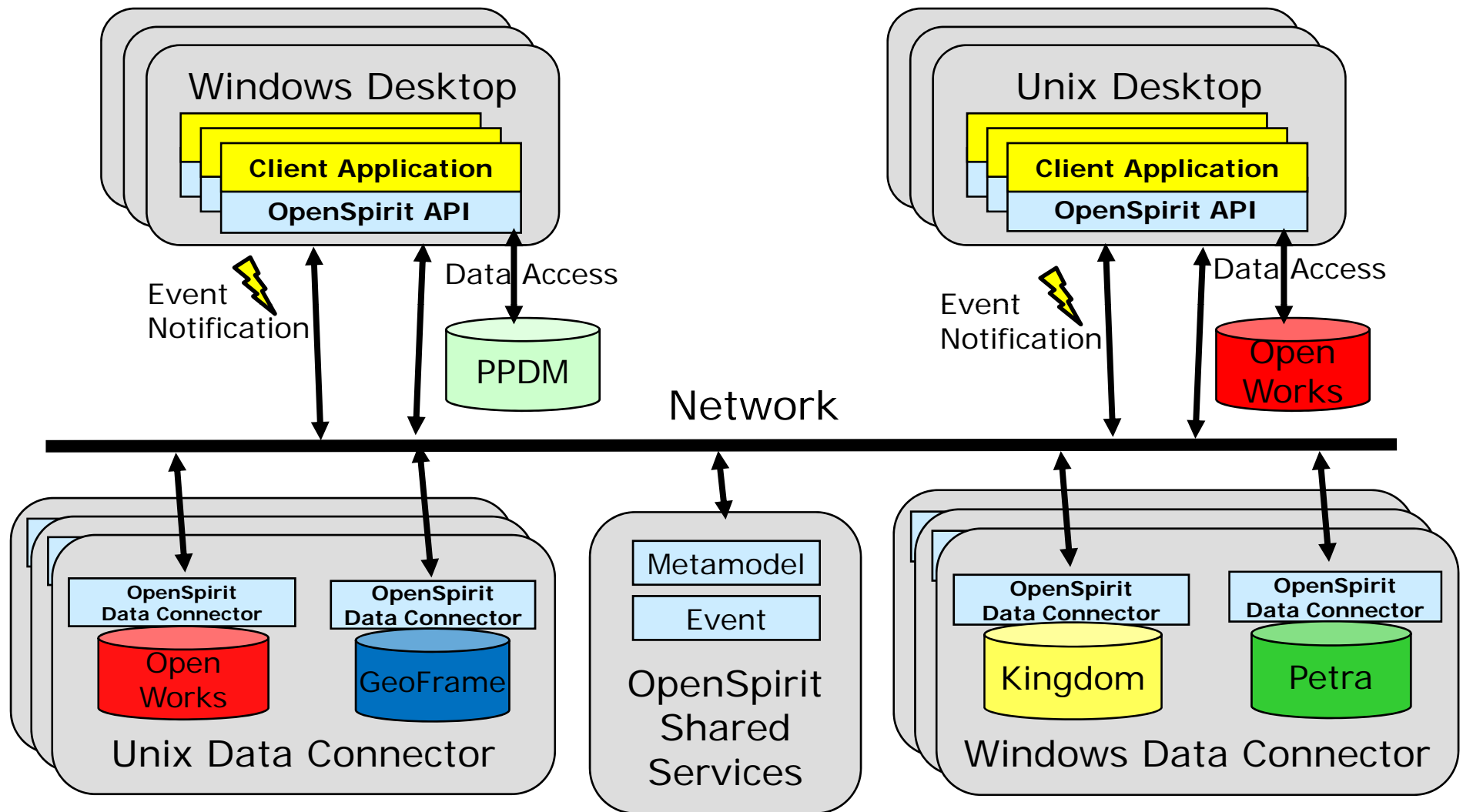
v3- Service Oriented Middleware

v2 APIs *Java, .NET, and C++ (Windows, Linux, and Solaris)*

v3 APIs *Java, .NET, and C++ (Windows, Linux, and Solaris)*



OpenSpirit v3 – A Process View

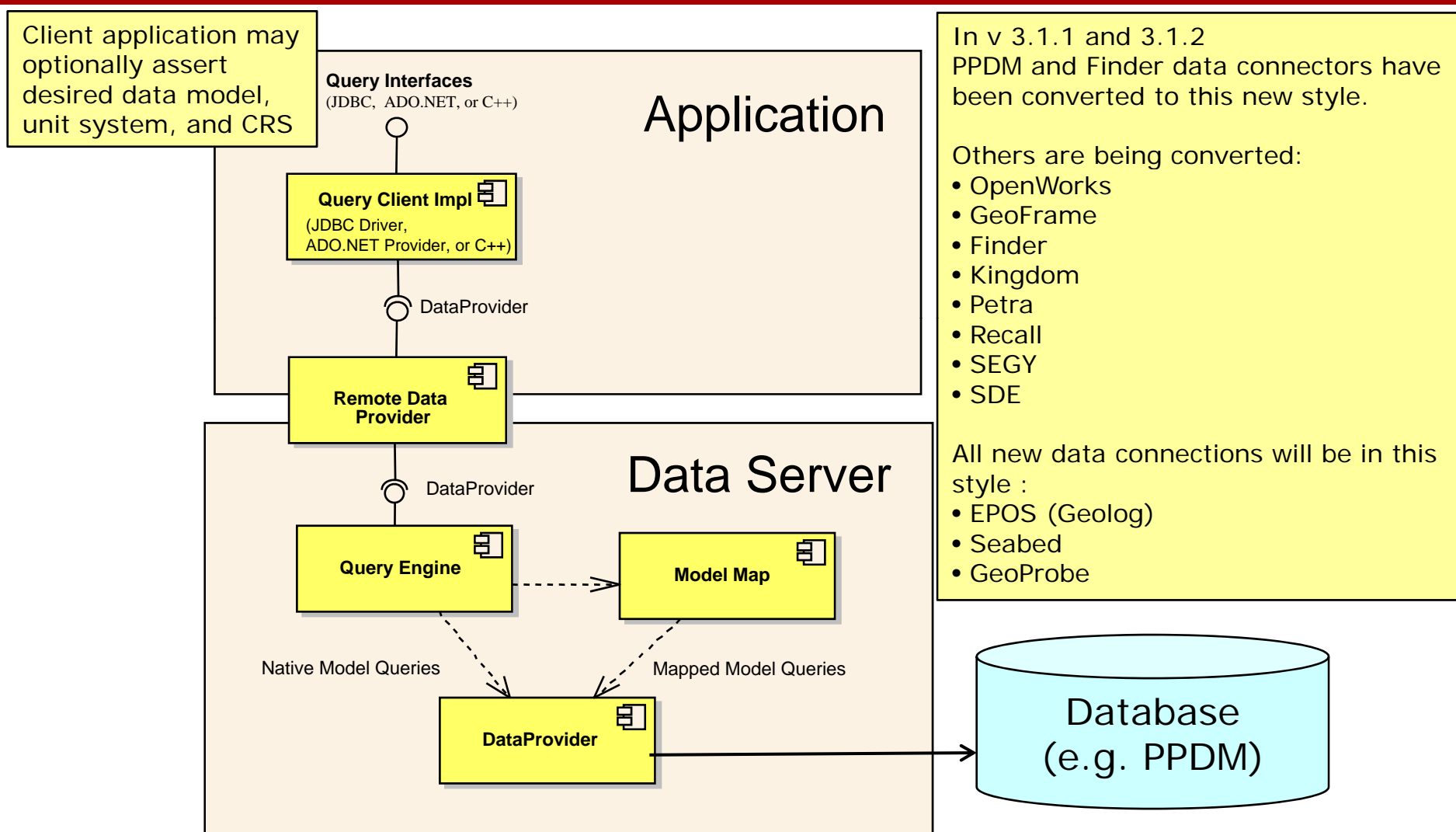


Developer Resources

- Dev Kit installer downloadable from developer forum (<http://devnet.openspirit.com>) for registered users. Available on Linux, Solaris, and Windows
- Installs
 - OpenSpirit data model documentation
 - Language Specific
 - Required libraries
 - Example application (binary and source code)
 - API documentation
- Post questions (and review other answers) on developer forum
- Some example data available (e.g. Managed SEGY or PPDM Oracle backup) – more coming



New Data Access Pipeline



Data Service

- Discover data sources and their capabilities
- Remote data connectors activated on demand
- High performance & scalable queries
- Query API is true JDBC & ADO.NET
- Automatic unit and coordinate system conversions can be asserted
- Query scope may span multiple data sources – a single result set is returned
- Support for multiple data models
- Query language is true SQL with spatial extensions
 - Support for OGIS based geometry types (Point, LineString, Polygon, ...) and spatial constraints (Contains, Crosses, Disjoint, Equals, Intersects, Overlaps, Touches, Within)



Metamodel Service

- Discover entities, attributes and relationships of any known data model (meta model stored in Metadata Repository)
- May be used in conjunction with Data Service to design generic, data model independent , applications (e.g. generic data browser or reporting tool)



Bulk Data APIs

- Provide simple access to “bulk data”
- 3D Seismic
 - Trace Accessor – read/write specific traces
 - Plane Accessor – read/write inline, crossline, or timeslice planes
 - Sub-Volume Accessor – read/write sample “bricks”
- 2D Seismic
 - Trace Set Accessor – read/write specific traces
- Horizon Grid
 - 2D Grid Accessor – row/column range
 - Row/Column Accessor
- Integration of query service with bulk accessors, no need for entity objects



Event Service

- Simple event listener pattern
- Java & .NET thread pool based dispatching
- C++ thread pool or UI event loop dispatching
- User scoped events
 - Data Selection
 - Area of Interest
 - Point of Interest
 - Cursor Location
 - GIS Feature Selection
 - Coming- View Definition
- Extendable message content
- Global events
 - Data Change

Optionally assert
desired CRS



Unit Service

- 900+ POSC based unit definitions
- Performs unit conversions
- OpenSpirit v3 introduces:
 - Multiple catalog support (POSC units, GeoFrame units, OpenWorks units, ...)
 - Hierarchical unit categorizations (e.g. length, vertical length)
 - Multiple unit systems within catalogs (UnitMeasure, Unit associations)
 - Metadata driven catalogs
 - Cross catalog conversions
 - Query API supports quantity data types (unit & value)
 - In-process local service



Coordinate Service

- Supports Projected, Geographic, and Engineering (e.g. Seismic Bin/Grid) Systems
- Factories for finding or creating systems
- Performs coordinate transformations & datum conversions
- Used by query service to perform automatic transformations
- OpenSpirit v3 introduces:
 - EPSG metadata
 - ESRI projection engine replaces Mentor
 - 2D coordinate system support added
 - “Early” & “late” datum conversion binding to systems
 - In-process local service



OpenSpirit Data Model

- Well, Seismic, Interpretation, Culture, and Drilling entities
- Supports inheritance and relationships
- [OpenSpirit v2 Model](#)
- OpenSpirit v3 adds support for:
 - Native data models
 - Multiple application models



Integration of Your Application

- What style of OpenSpirit integration?
 - Import/export?
 - “Live” link?
- How to find OpenSpirit objects?
 - Listen for data selection events?
 - Use your own GUI to control queries?
- How to deal with units and coordinates?
 - Map your units, CRS designations to OpenSpirit?
 - Use the session’s preferred units and CRS?
- What objects to use?
- What events to use?
- OpenSpirit guest developer program can help you make these design decisions



Example Client Application



Connect to OpenSpirit and listen for wellbore selection events. When one is received, print out the wellbore name. TD and bottom location in desired units and CRS



C# Example

```
using System;
```

```
{
```

```
    class SimpleListener
```

```
    {
```

```
        OpenSpirit.IOpenSpirit osp;
```

```
        static void Main(string[] args)
```

```
        {
```

```
            SimpleListener listener = new SimpleListener();
```

```
            listener.InitializeOpenSpirit();
```

```
            System.Console.ReadLine();
```

```
        }
```

```
        void InitializeOpenSpirit()
```

```
        {
```

```
            osp = OpenSpirit.OpenSpiritFactory.CreateOpenSpirit();
```

```
            osp.Connect("MyCompany.MyFirstApp");
```

```
            OpenSpirit.Event.DataSelectionEventHandler eventHandler = new
```

```
                OpenSpirit.Event.DataSelectionEventHandler(OnReceivedDataEvent);
```

```
            OpenSpirit.Event.IDataSelectionEventPortal eventPortal = osp.EventService.GetDataSelectionEventPortal(null);
```

```
            OpenSpirit.Metamodel.IEntityDefinition[] entities = {
```

```
                osp.MetamodelService.GetModel(null,null).GetEntity("EpiWell_Wellbore") };
```

```
            eventPortal.AddDataSelectionEventHandler(eventHandler, null, entities);
```

```
            eventPortal.ListeningEnabled = true;
```

```
        }
```

```
        void OnReceivedDataEvent(object sender, OpenSpirit.Event.DataSelectionEventArgs args)
```

```
        {
```

```
            QueryFromKeys(args.Keys);
```

```
        }
```

Create instance of class and listen until console input is received

Connect to OpenSpirit and check out UAA license

Register to receive data selections for wellbores (from default OpenSpirit data model) and begin listening

Event callback then calls method to issue query for selected wellbores



C# Example – cont.

```
void QueryFromKeys(OpenSpirit.Data.IDataKey[] keys)
```

Set query scope based on received keys

Assert metric unit system and WGS-84 CRS for query context

```
{
    OpenSpirit.Data.IQueryScope scope = osp.DataService.QueryFactory.CreateQueryScope(keys);
    OpenSpirit.Unit.IUnitSystem unitSystem = osp.UnitService.GetCatalog("POSC", null).GetSystem("Metric");
    OpenSpirit.Carto.ICoordinateReferenceSystem crs = osp.CoordinateService.FindSystemByCode(
        OpenSpirit.Carto.CommonEPSGCodes.WORLD_GEODETTIC_SYSTEM_1984_2D);
    OpenSpirit.Data.IQueryContext context = osp.DataService.QueryFactory.CreateQueryContext(unitSystem, crs, null);
    OpenSpirit.Data.OspDbConnection dbConnection = osp.DataService.GetOspConnection(scope, context);
    OpenSpirit.Data.OspDbCommand command = dbConnection.CreateOspCommand(null);
    command.CommandText = "SELECT name, totaldepth, bottomlocation from EpiWell_Wellbore where PrimaryKey$ in (?)";
    command.Parameters.Add(command.CreateParameter());
    command.Parameters[0].Value = keys;
    OpenSpirit.Data.OspDbDataReader reader = command.ExecuteOspQuery();
    while (reader.Read())
    {
        Console.WriteLine( reader.GetString(0) + " " + reader.GetFloatQuantity(1) + " " + reader.GetGeometry(2) );
    }
}
```

Execute SELECT query to query for name, total depth, and bottom location for received bores

Loop over returned rows and write to console (use ToString() method of OpenSpirit datatypes)

Note: Real application should use try/catch blocks and release resources!

Next Steps

Better understand how you can use OpenSpirit

- Initiate a free evaluation in your office
- Schedule time in our “guest developers” program in our office



For More Information

- info@openspirit.com
- <http://www.openspirit.com>
- <http://devnet.openspirit.com>
(available to dev kit customers)



Thank you!

Clay Harter, Chief Technology Officer
+1 281 295 1407
clay.harter@openspirit.com

