Microsoft®
tech·ed
Russia | 2011

9–10 НОЯБРЯ 2011
МОСКВА

Microsoft tech·ed Russia 2011

9–10 НОЯБРЯ 2011 · МОСКВА

ARC202

Providing Broad Access to Data with OData

David Chappell
Principal
Chappell & Associates

Agenda

- Describing OData: The Basics
- Examining OData: A Closer Look at the Technology

Describing OData: The Basics

The Problem: Accessing Diverse Data

- Many kinds of data sources exist, such as:
  - Custom apps with relational storage
  - Cloud platforms with relational storage, NOSQL storage, etc.
  - Content management systems with lists, etc.
- Many kinds of clients can use data, such as:
  - Web browsers
  - Phone/tablet apps
  - Business intelligence (BI) tools
  - Custom apps
- How can all of these clients access all of these data sources?

A Solution: OData

- The Open Data Protocol (OData) addresses this problem
  - It lets diverse clients access diverse data sources
- OData defines:
  - An abstract data model
  - A standard access protocol
    - With multiple data serialization formats
- OData is a generic technology
  - It's not tied to Windows
  - Although Microsoft provides tools and libraries for implementing OData

Illustrating OData

Clients

- Web Browsers (Internet Explorer, Firefox, …)
- Mobile Phones (Android, iPhone, Windows Phone 7)
- BI Tools (Microsoft Excel, Tableau Desktop)
- Custom Applications (.NET, Java, PHP, …)

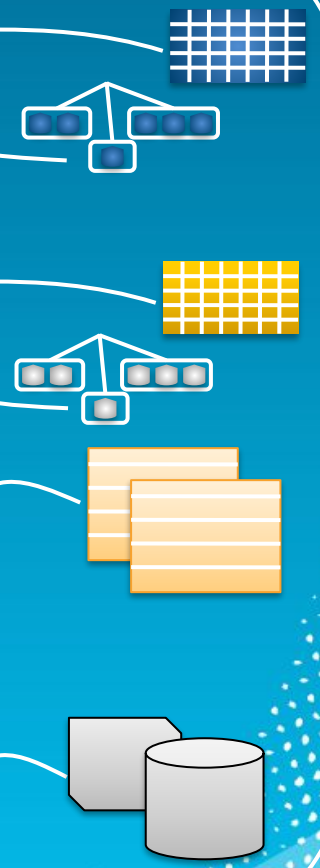
. . .

OData

Data Sources

- Custom Applications (.NET, Java)
- Cloud Storage (Windows Azure Tables, SQL Azure)
- Content Management (SharePoint 2010, Webnodes)
- Windows Azure Marketplace Data

. . .



OData Components (1)

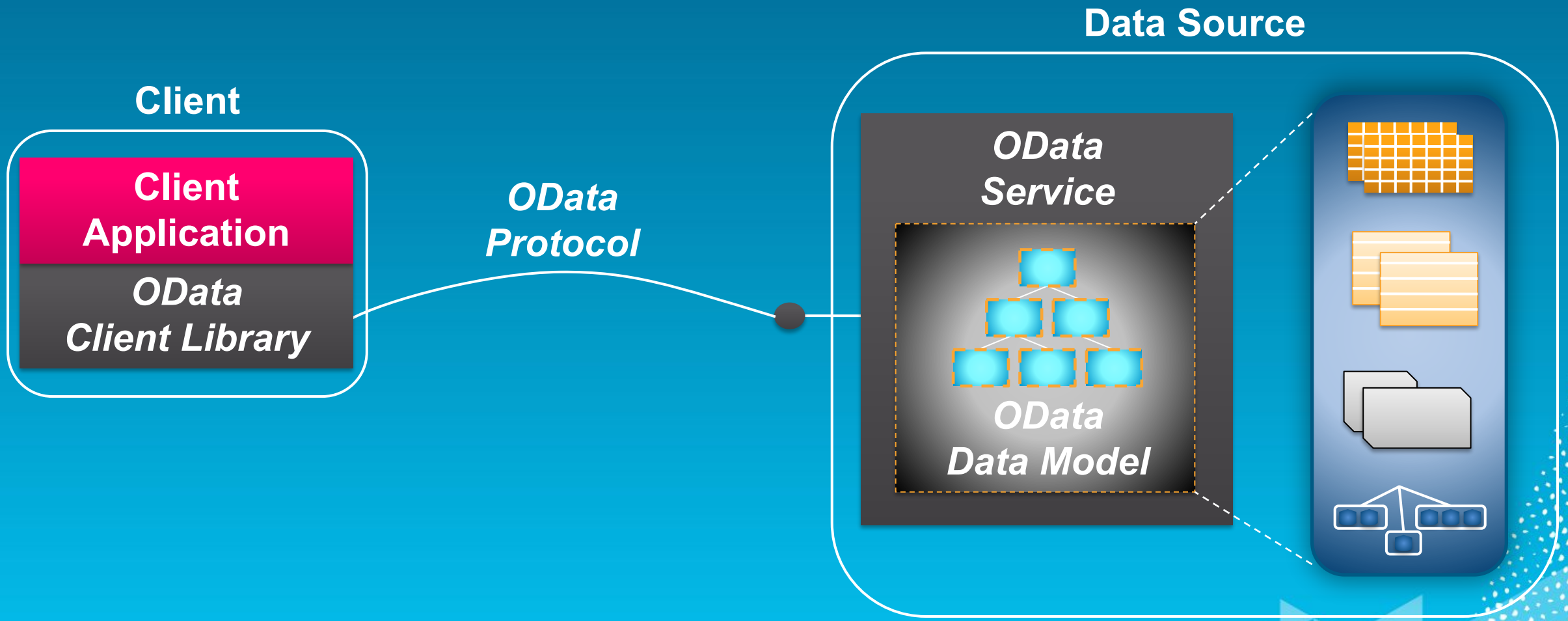
- The *OData data model*
  - Provides a generic way to organize and describe data
  - Uses the Entity Data Model (EDM) from Microsoft's Entity Framework
- The *OData protocol*
  - Based on HTTP
  - Lets a client make requests to and get responses from an OData service

OData Components (2)

- An *OData service*
  - Exposes an endpoint that allows access to data
  - Uses the abstractions of the OData data model to translate data between its underlying form (e.g., relational tables) into the format sent to the client
- *OData client libraries*
  - Make it easier to create software that accesses data via the OData protocol

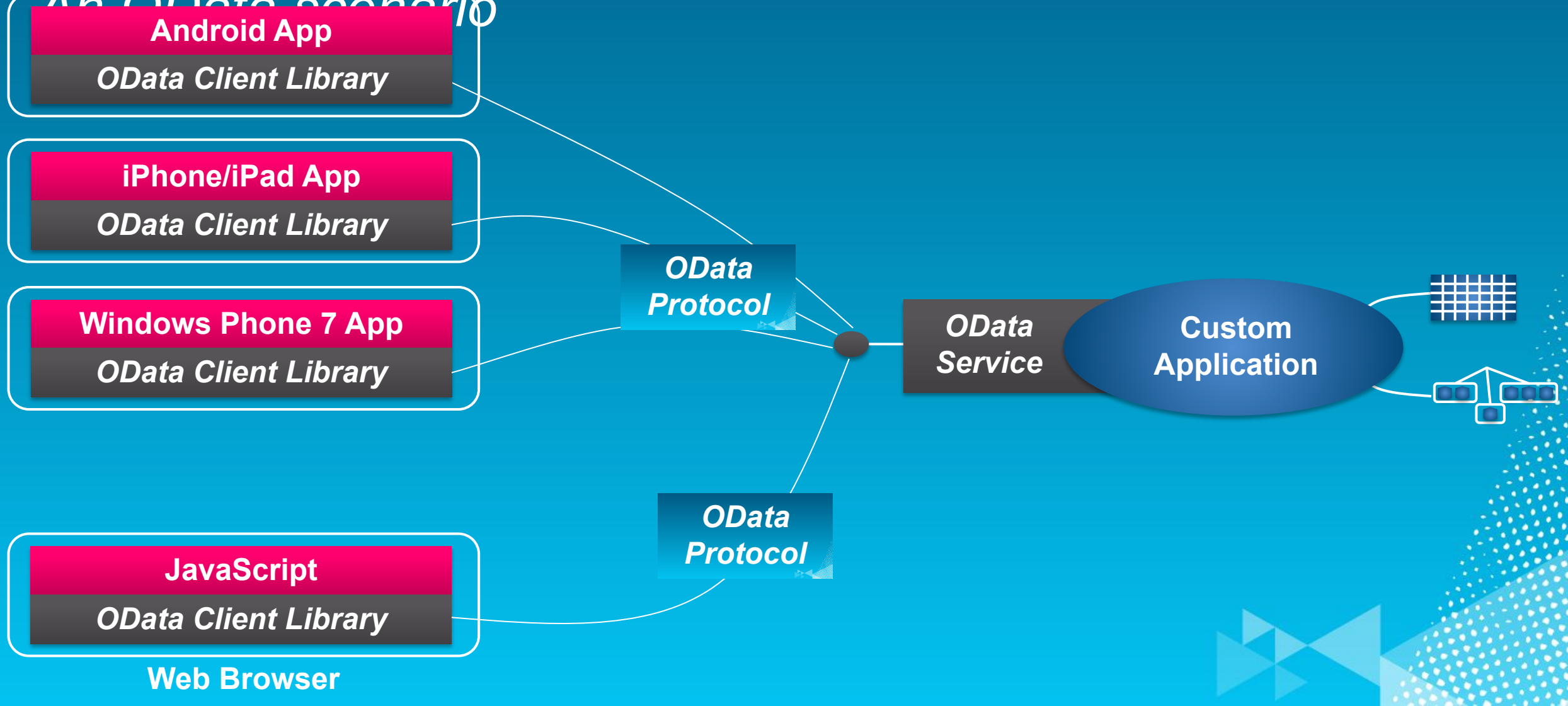
OData Components

An illustration



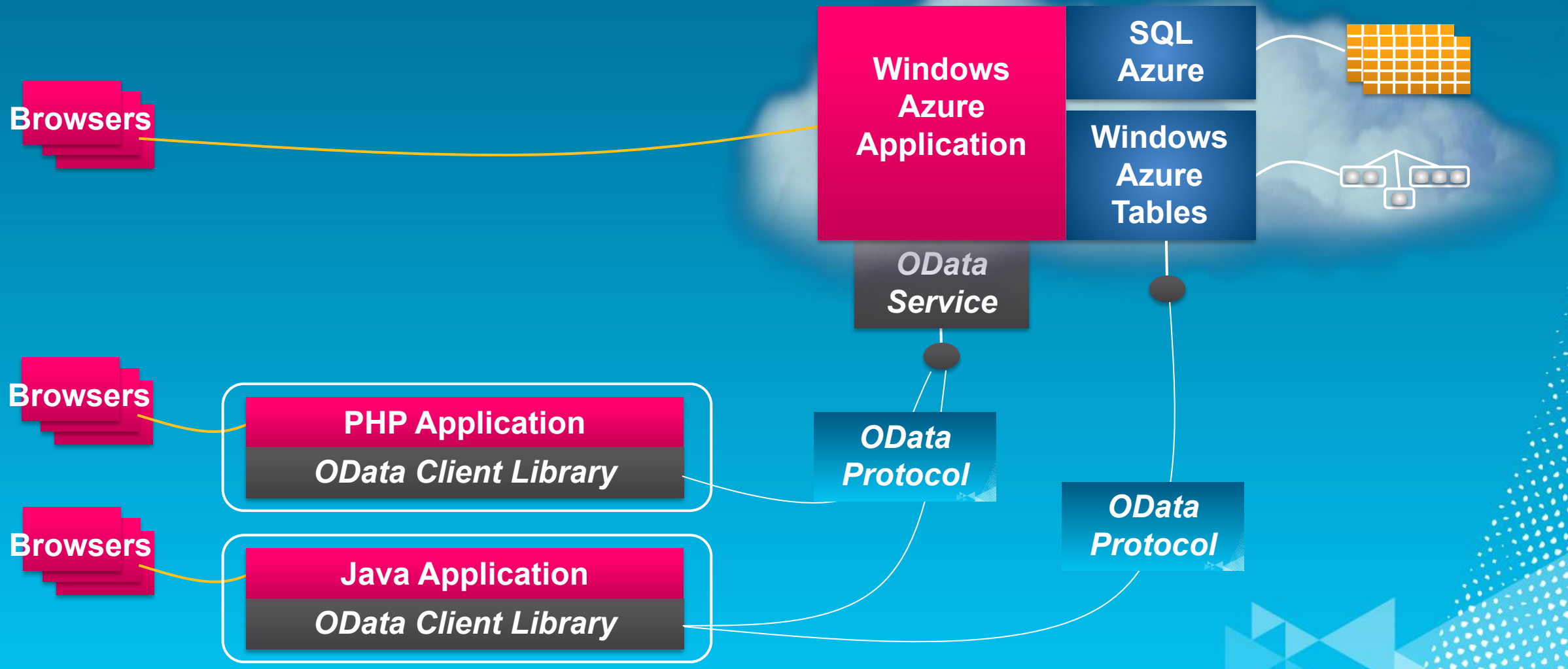
Accessing Data from Mobile Devices and Browsers

An OData scenario

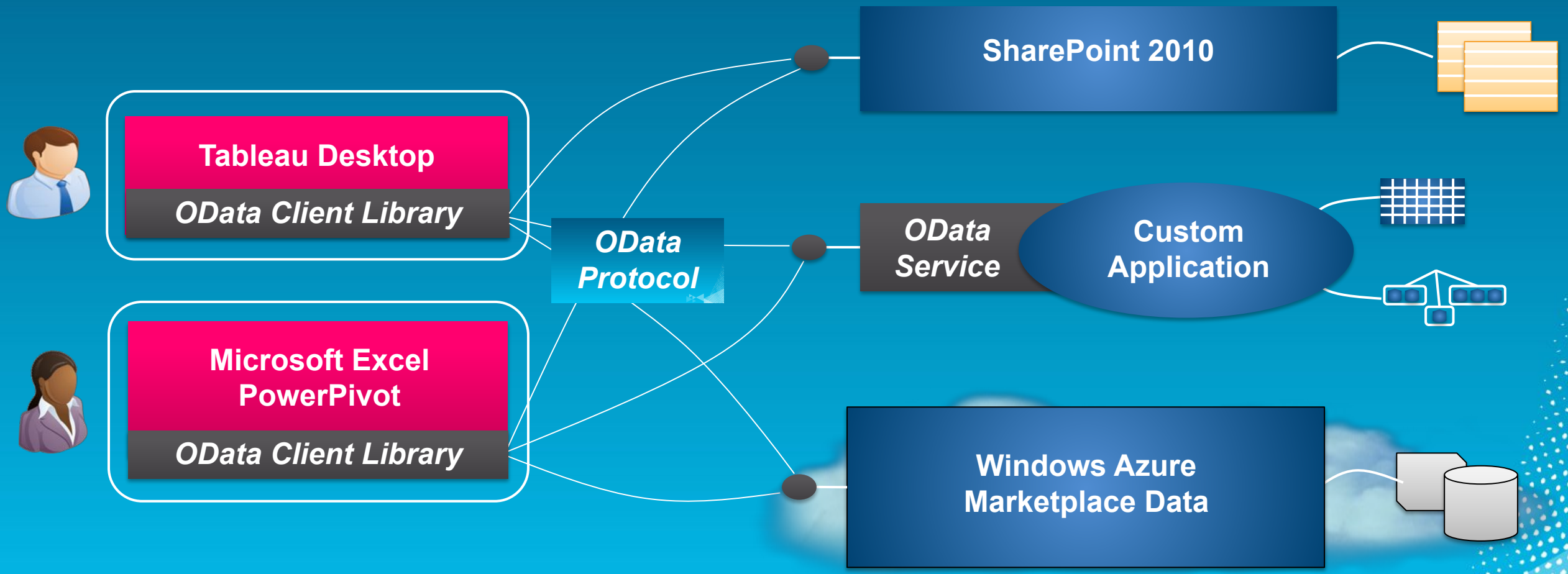


Exposing Data from a Cloud Application

An OData scenario



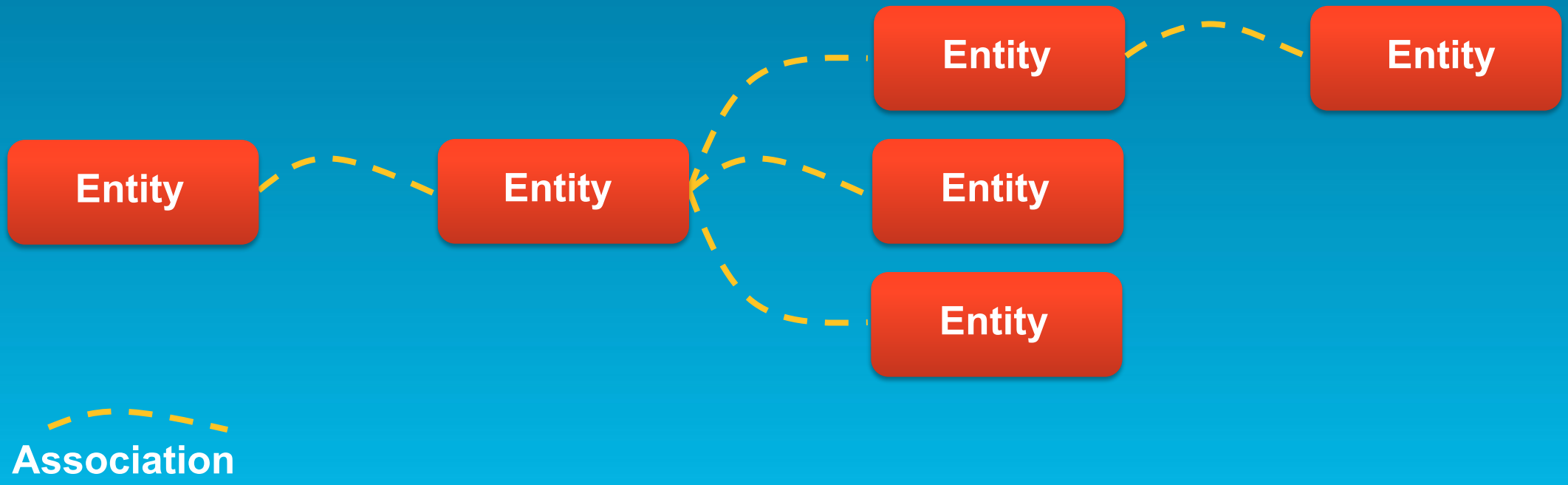
Using Diverse Data Sources with BI Tools
An OData scenario



Examining OData: A Closer Look at the Technology

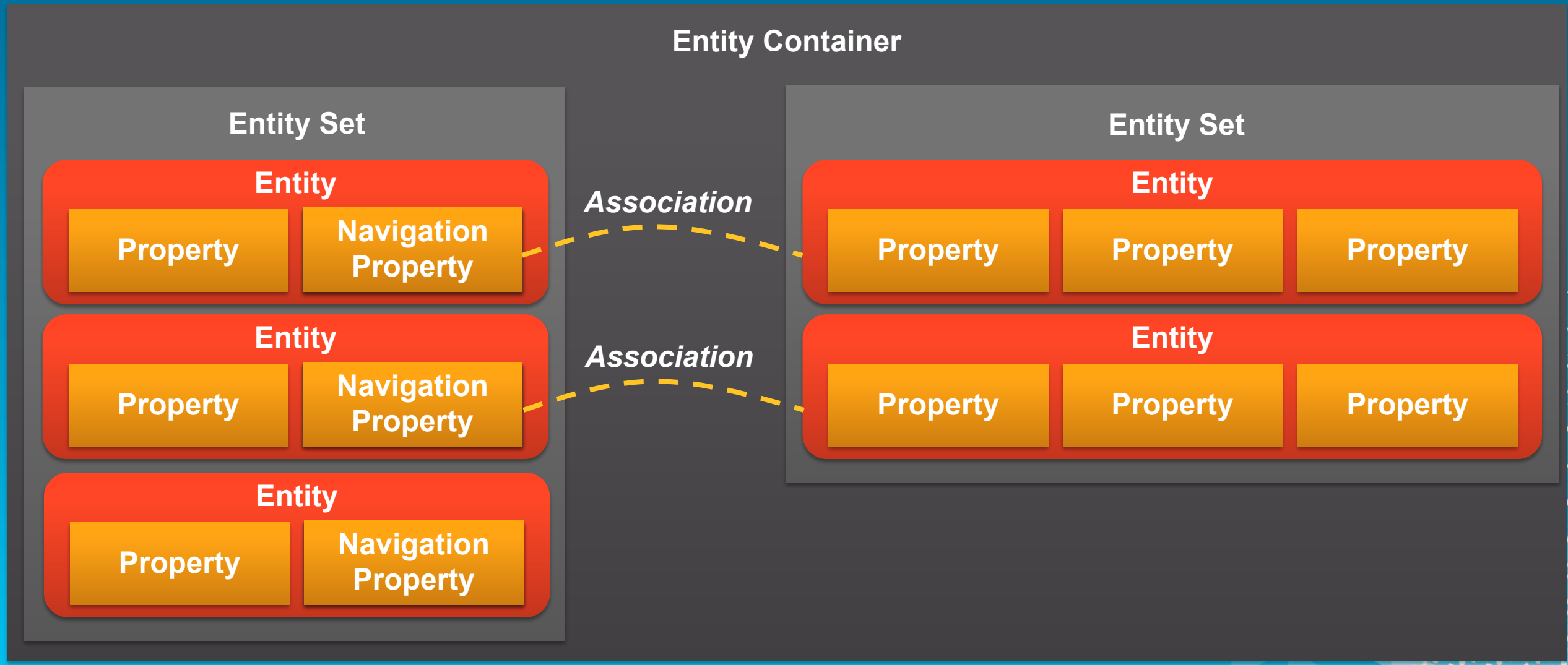
The OData Data Model

Entities and associations in the EDM



The OData Data Model

A closer look at the EDM



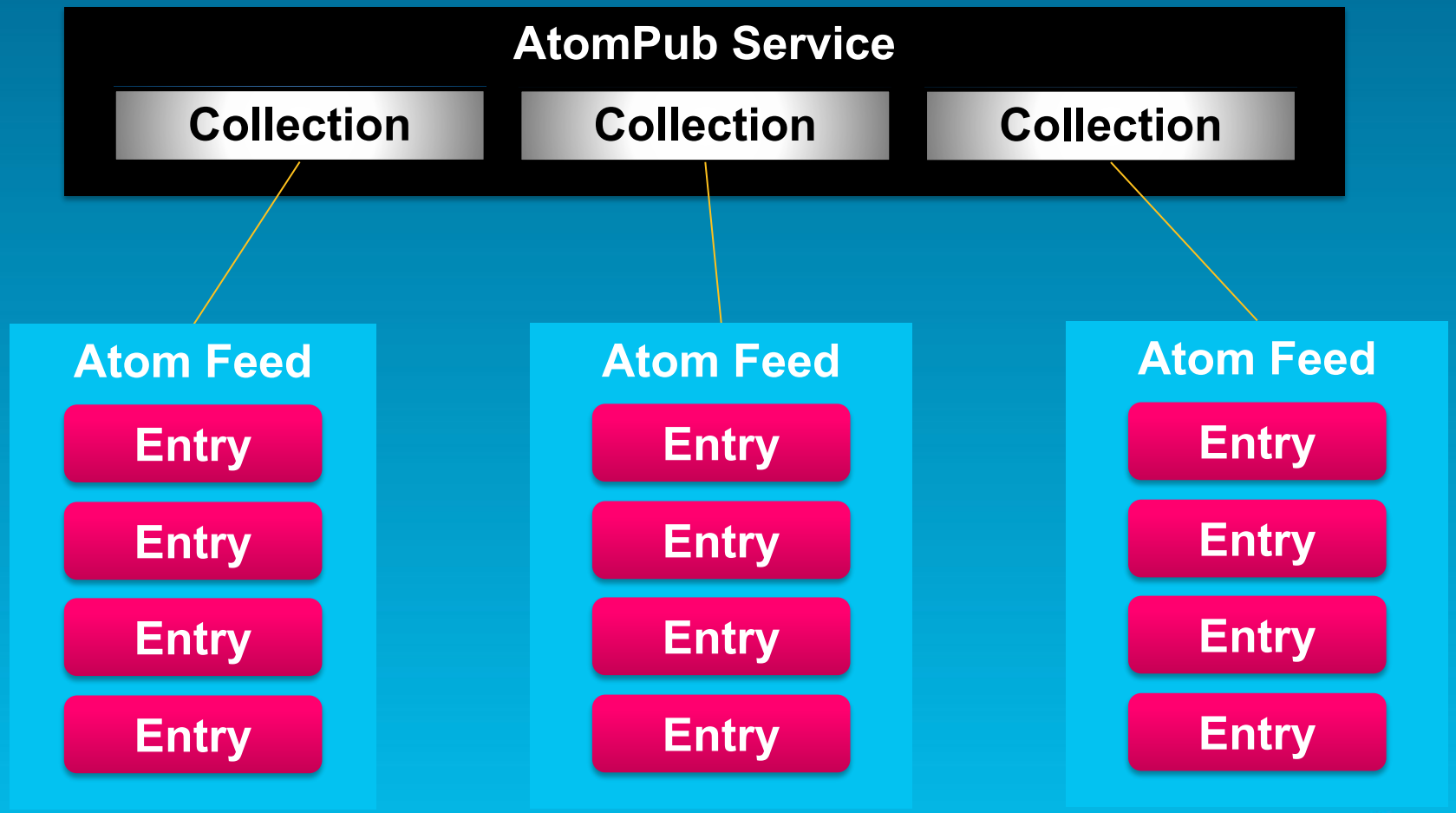
The OData Protocol

Basics

- Based on HTTP
- Example verbs:
  - POST: Creates a new entity
  - GET: Reads data from one or more entities
  - PUT: Updates an existing entity, replacing all of its properties
  - DELETE: Removes an entity
  - MERGE: Updates an existing entity, but replaces only specified properties
- Supported serialization options:
  - XML-based Atom/AtomPub
  - JavaScript Object Notation (JSON)

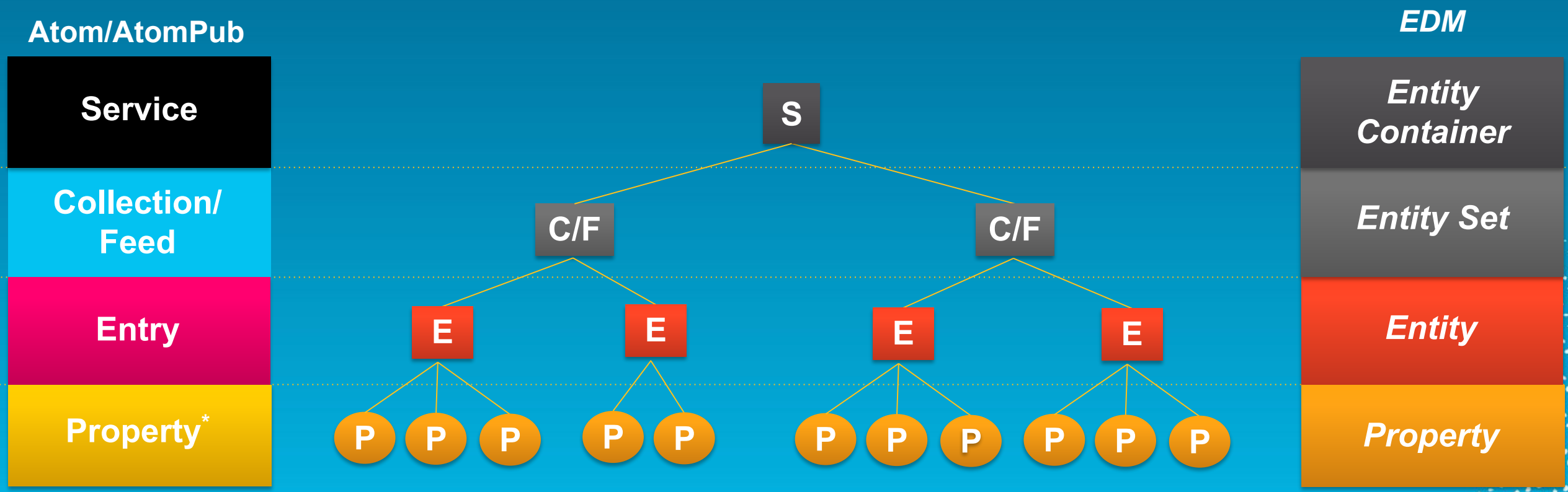
The OData Protocol

Illustrating Atom and AtomPub



The OData Protocol

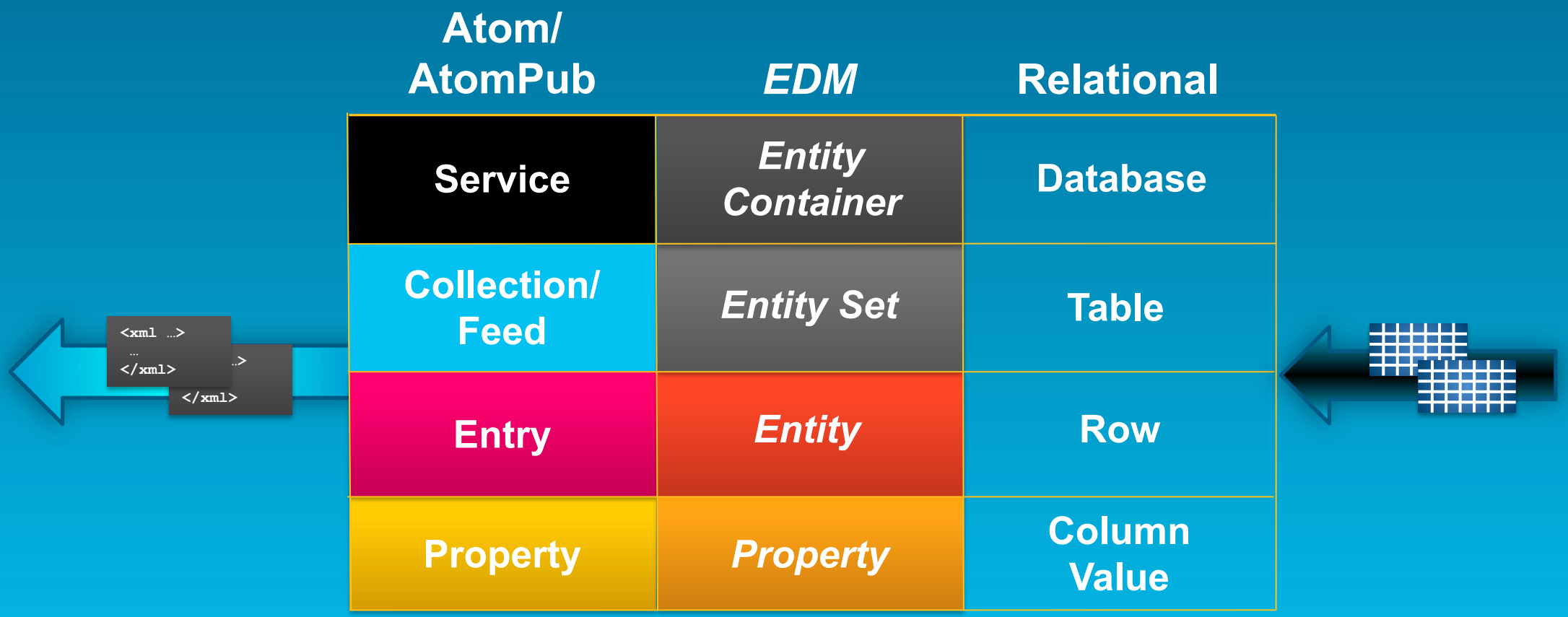
Mapping between EDM and Atom/AtomPub



*OData-defined extension to Atom

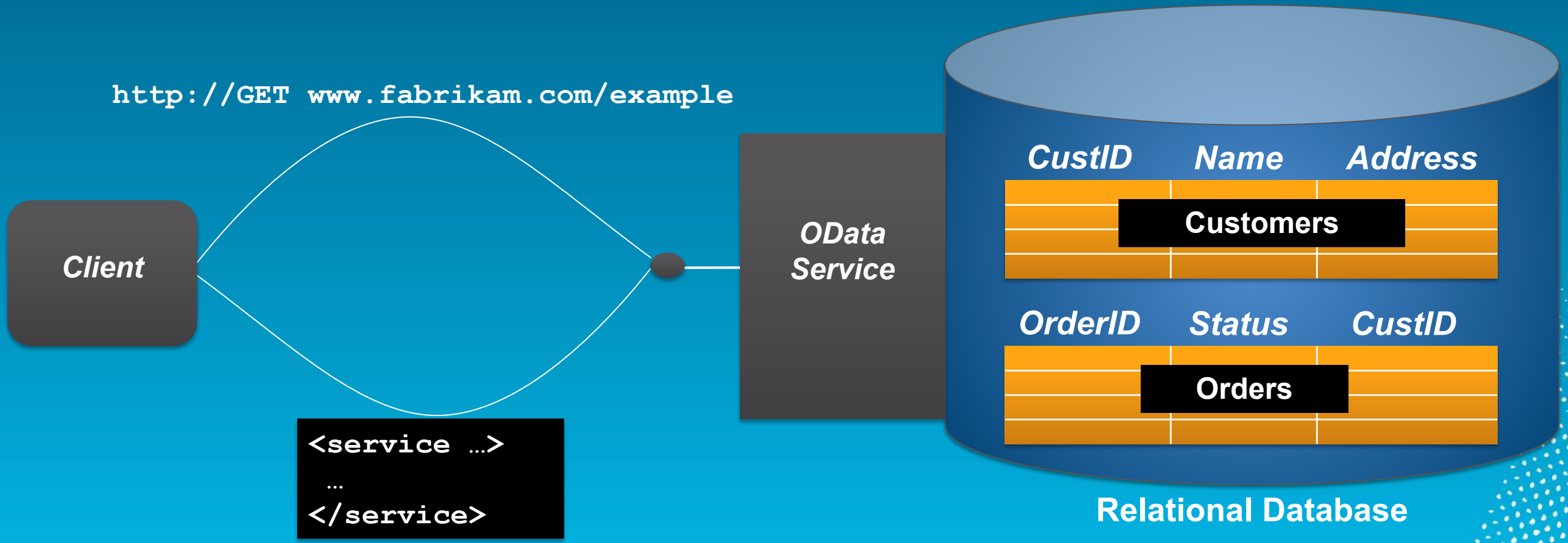
The OData Protocol

Serializing relational data in Atom/AtomPub



The OData Protocol

Getting an AtomPub service document



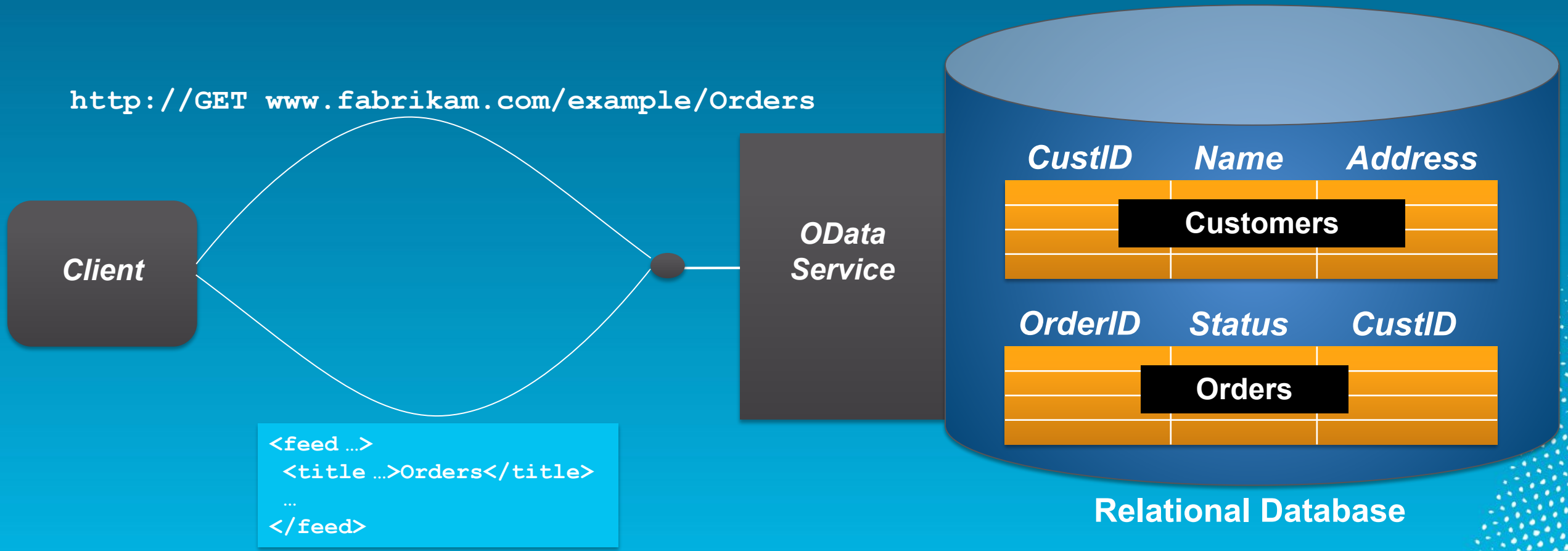
The OData Protocol

An example AtomPub service document

```
<service …>
…
  <collection href="Customers">
    <atom:title>Customers</atom:title>
  </collection>
  <collection href="Orders">
    <atom:title>Orders</atom:title>
  </collection>
…
</service>
```

The OData Protocol

Getting an Atom feed document



The OData Protocol

An example Atom feed document (1)

```
<feed xmlns:m= … xmlns:d= … >
  <title type="text">Orders</title>
    …
  <entry>
    …
    <content type="application/xml">
      <m:properties>
        <d:OrderID m:type="Edm.Int32">3501</OrderID>
        <d:Status>Shipped</Status>
        <d:CustID m:type="Edm.Int32">867734</CustID>
      </m:properties>
    </content>
  </entry>
```

An OData-defined extension to Atom/AtomPub

An Entity Data Model-defined data type

The OData Protocol

An example Atom feed document (2)

```
<entry>
  …
  <content type="application/xml">
    <m:properties>
      <d:OrderID m:type="Edm.Int32">5630</OrderID>
      <d:Status>Placed</Status>
      <d:CustID m:type="Edm.Int32">8499734</CustID>
    </m:properties>
  </content>
</entry>
<entry>
  …
</entry>
…
</feed>
```

The OData Protocol

Some other options

- Returning the number of entries in a feed:

```
http://GET www.fabrikam.com/example/Orders/$count
```

- Returning only the entry whose primary key is 5630:

```
http://GET www.fabrikam.com/example/Orders(5630)
```

The OData Protocol

Serializing data with JSON

- The first request:

```
http://GET www.fabrikam.com/example
```

- The response:

```
{ "d" : {
   "EntitySets": ["Customers", "Orders"]
} }
```

The OData Protocol

Serializing data with JSON

- Requesting data:

```
GET http://www.fabrikam.com/example/Orders(5630)
```

- The response:

```
{ "d" : {
   "results": {
     "OrderID": 5630,
     "Status": "Placed",
     "CustID": 8499734 }
} }
```

The OData Protocol

Issuing queries (1)

- The OData query language is expressed as options appended to a GET request

- Examples:
  - **`$top=n`**: Returns only the first *n* entities in an entity set
    - That is, the first *n* entries in an Atom feed
  - **`$skip=n`**: Skips the first *n* entities in an entity set
    - Lets a client retrieve a series of distinct pages on subsequent requests
  - **`$format`**: Determines whether data should be returned in JSON or Atom/AtomPub
    - Default is Atom/AtomPub

The OData Protocol

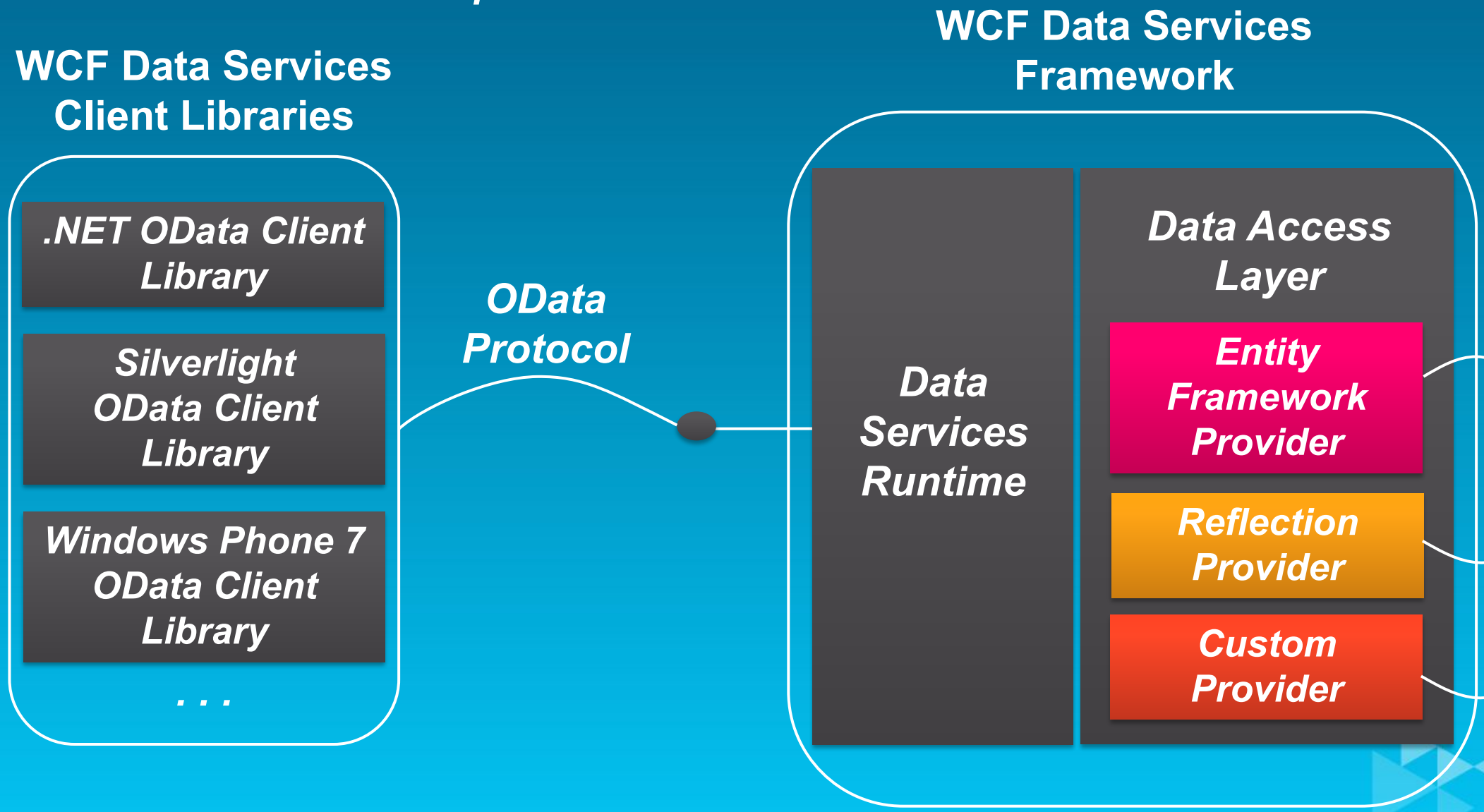
Issuing queries (2)

- Expressions:
  - **`$orderby=<expression>`**: Orders results by the value of one or more properties in those results
  - **`$filter=<expression>`**: Returns only entities that match the specified expression
  - **`$select=<expression>`**: Returns only the specified properties in an entity

- Example:

```
http://GET www.fabrikam.com/example/Orders?$filter=OrderID
gt 3000 and OrderID lt 5000
```

Implementing OData
What Microsoft provides



Conclusion

- OData provides broad support for accessing many kinds of data
- It includes:
  - A general data model
  - A protocol
    - With serialization in Atom/AtomPub or JSON
- Microsoft also provide support for implementing OData clients and services

- OData addresses an important problem

Feedback

Dear attendees!

Your feedback is appreciated!

In notepad that can be found in attendee’s infopack, you will find a feedback form

Please, rate the session and pass the feedback form to the moderator on exit

To take part in a lottery, please do not forget to mark the feedback form with the number of your badge

Thank you!

Questions?

- Providing Broad Access to Data with OData, ARC202
- David Chappell
  - Principal, Chappell & Associates
  - david@davidchappell.com
  - www.davidchappell.com

- You can ask your questions at Microsoft zone in Hall 17 within an hour after end of this session

For Further Reading

- Introducing OData

  http://www.davidchappell.com/writing/white_papers/Introducing_OData_v1.0--Chappell.pdf

About the Speaker

David Chappell is Principal of Chappell & Associates (www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technology. David has been the keynote speaker for more than a hundred events and conferences on five continents, and his seminars have been attended by tens of thousands of IT decision makers, architects, and developers in forty countries. His books have been published in a dozen languages and used regularly in courses at MIT, ETH Zurich, and other universities. In his consulting practice, he has helped clients such as Hewlett-Packard, IBM, Microsoft, Stanford University, and Target Corporation adopt new technologies, market new products, train their sales staffs, and create business plans. Earlier in his career, David wrote networking software, chaired a U.S. national standards working group, and played keyboards with the Peabody-award-winning Children's Radio Theater. He holds a B.S. in Economics and an M.S. in Computer Science, both from the University of Wisconsin-Madison.