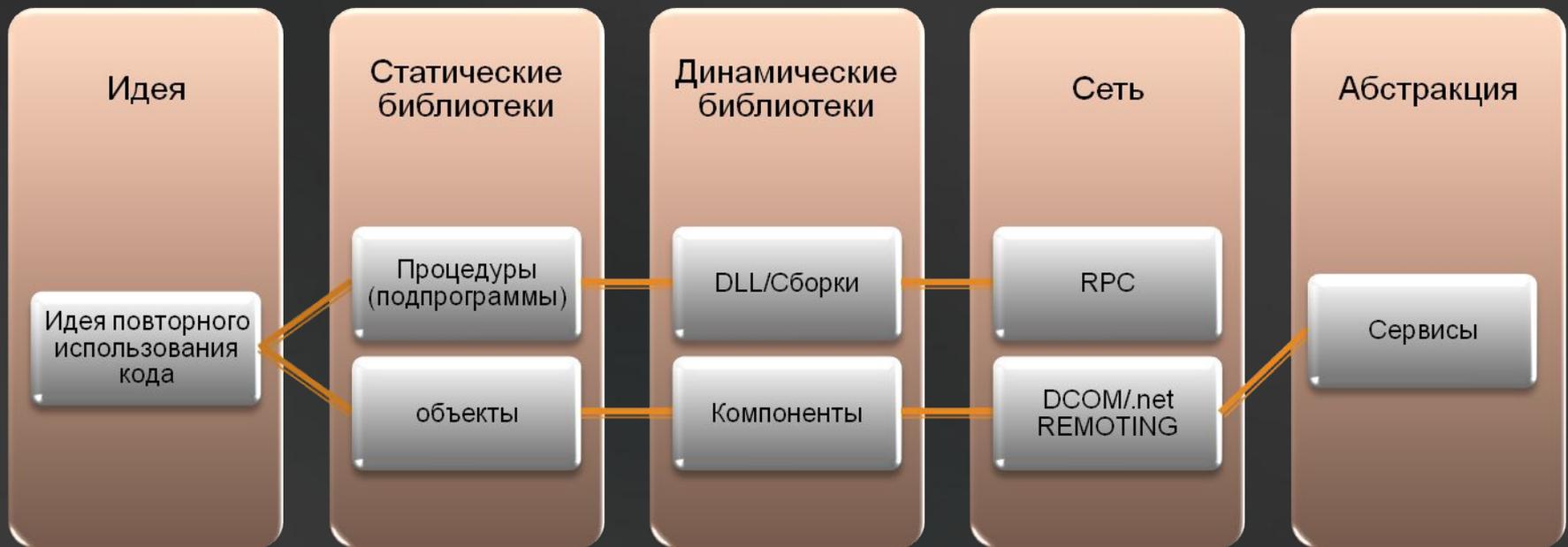


Turn your data into
a new form of energy.

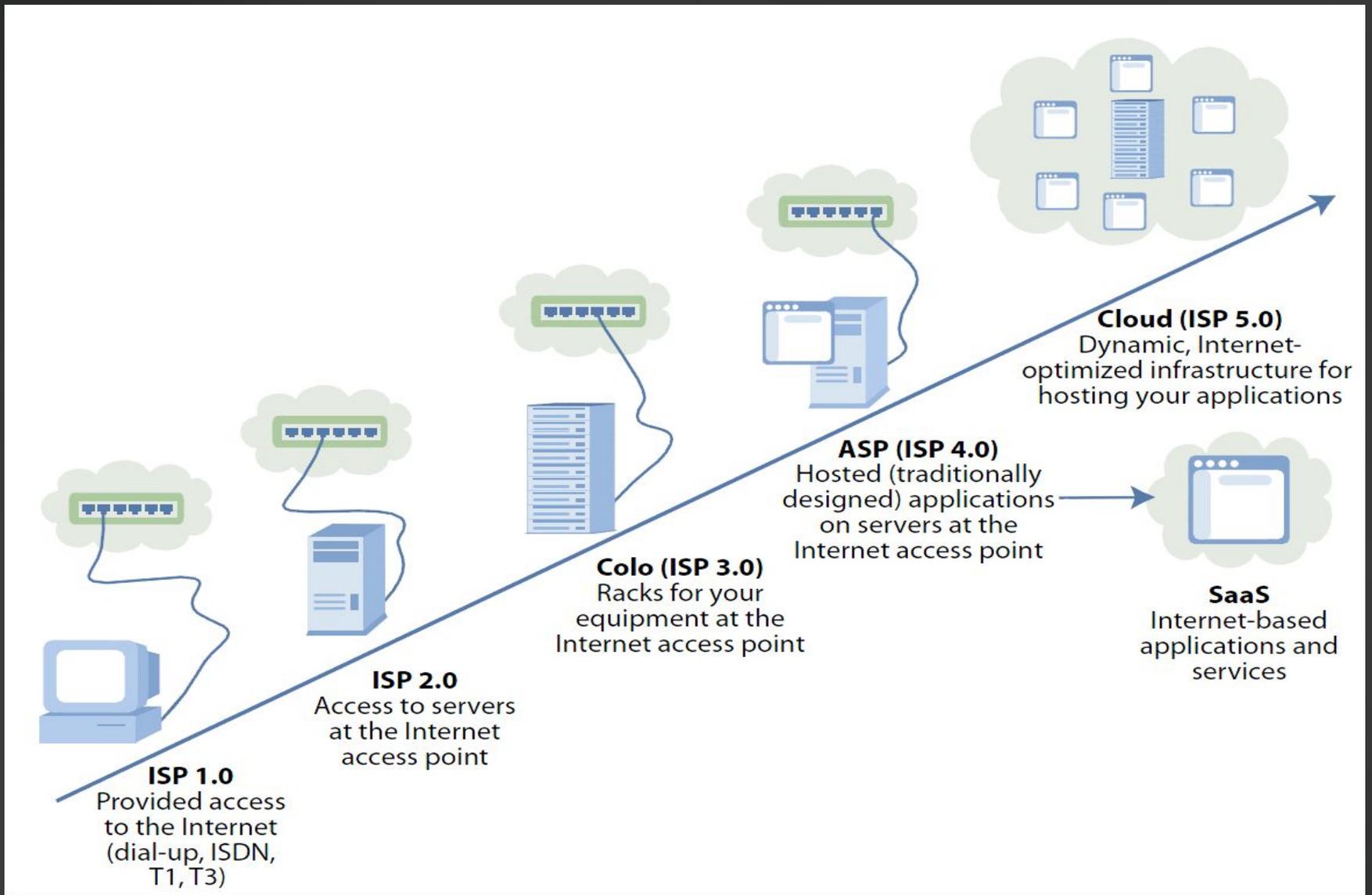
Введение в SQL Data Services

Алексей Шуленин
эксперт по технологиям обработки и анализа информации
департамент стратегических технологий
Российское представительство Microsoft
Microsoft Corporation

Развитие компонентного подхода



ЭВОЛЮЦИЯ ХОСТИНГА



Преимущества облачной модели с точки зрения заказчика

- Не требует крупных предварительных инвестиций

заказчика

В случае неудачи проекта не нужно ломать голову, куда пристроить дорогостоящее оборудование

- В случае удачи автоматически решается проблема масштабирования со 100 пользователей на 1000, на 10000, ...
- **Меньше операционных издержек**
 - Размещение громадных количеств пользователей на высокомасштабируемом сервере дает стоимость в расчете на пользователя на порядки меньше, чем в случае развертывания системы на предприятии
- **Пересекает границы организации**
 - Файрвол не преграда
 - Ценность информации многократно возрастает, если расширить ее на партнеров, поставщиков, клиентов
 - Проще, чем строить экстранет
- **Быстрая подготовка к работе**
 - Пользователям не требуется ждать внедрения и настройки специализированных средств

Три модели DaaS

- Multitenancy hosted model
 - Несколько организаций шарят одну базу
 - Грубо говоря, каждый экаунт привязан к своей схеме
 - Стандартный доступ через SOAP, REST, XML, ODBC, JDBC, SQL
 - Можно создавать таблички, DRI, запрашивать, добавлять, изменять данные
 - Нет задачи сопровождать и поддерживать
- Dedicated database hosted model
 - Каждой организации выделяется по БД
 - Все БД шарят общую инфраструктуру: сервера, сторидж, ...
 - Доступ по стандартным протоколам, что и в пред.случае
 - DBA клиента может апгрейдить, тюнить базу, вообще делать с ней, что хочет – она же его выделенная
- Database container model
 - Обращаемся уже не к таблицам, а к сущностям
 - Более высокий уровень абстракции
 - Сущности, вообще говоря, нереляционные, хотя основываются на реляционных таблицах
 - Базовая сущность – набор свойств со значениями, сущности могут

Платформа Windows Azure

 Windows Live

 Microsoft Office Live

 Microsoft Dynamics CRM Online

 Microsoft SharePoint Online

Azure Services Platform

 Live Services

 Microsoft .NET Services

 Microsoft SQL Services

Microsoft Dynamics CRM Services

Microsoft SharePoint Services

 Windows Azure™

Задачи и решения

- Транспорт и инфраструктура сервисов
 - Windows Communication Foundation
 - Масштабируемый хостинг сервисов и приложений
 - Windows Azure
 - Связь разнородных сервисов между собой
 - .NET Services
 - Авторизация
 - LiveID, CardSpace (Client), .Net Services
 - Масштабируемая свободно структурированная база данных, доступная по REST и SOAP
 - SQL Services
 - Структурированная база данных в виде REST
 - ADO.NET Data Services
 - Синхронизация данных
 - ADO.NET Synchronization Services, Sync Framework, Live Framework

Характеристики SDS



Масштабируемость

- Миллионы геораспределенных машин, распределяемая по географическому принципу доставка информации петабайты данных, петафлопы вычислительной мощности

Уменьшение капиталозатрат

- Недорогие процессоры, диски IDE/SATA

Уменьшение операционных издержек

- Самомониторинг, самонастройка, самоизлечение

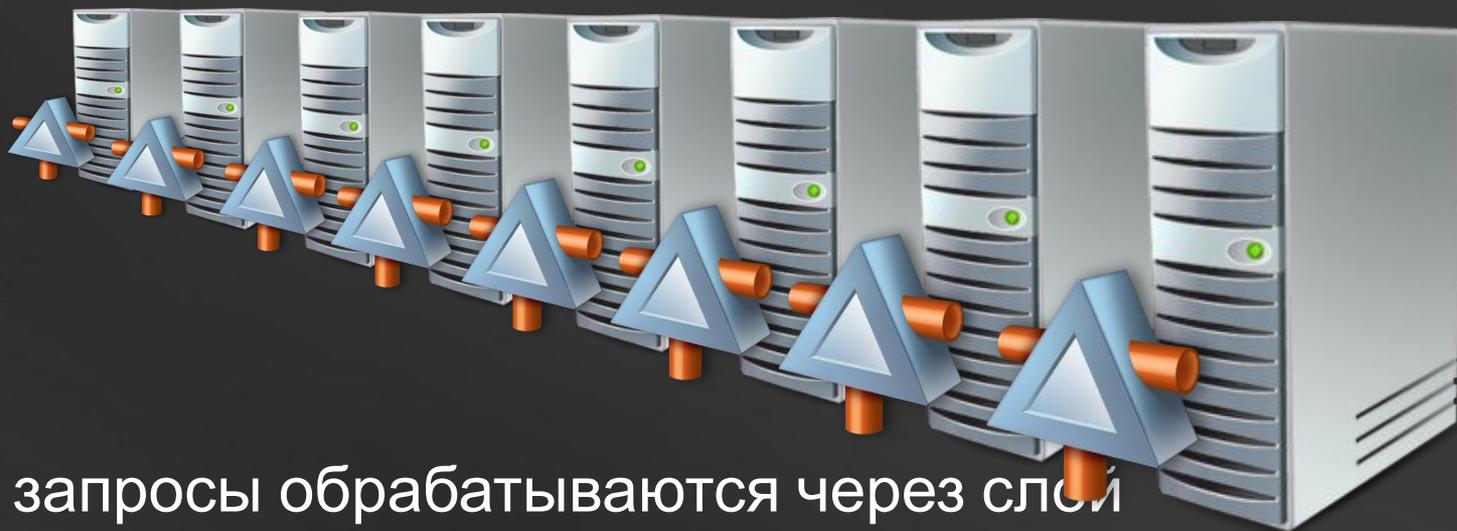
Предсказуемое детерминированное последовательное поведение

- На которое не оказывают влияния программные, аппаратные и коммуникационные сбои

Высокая производительность

- Инновационные алгоритмы

Параллельная обработка



- Все запросы обрабатываются через слой SDS Front End
- Запрос, разбитый на части, будет адресован нескольким серверам



В открытом тестировании с сентября 2008 г.

Nomination Survey - Windows Internet Explorer
https://connect.microsoft.com/Survey/NominationSurvey.aspx?SurveyID=7044&ProgramID=2500&SiteID=681

Welcome, Leshik Sign out | Your Profile | Help
Choose one of Your Connections

HOME YOUR DASHBOARD CONNECTION DIRECTORY Search GO

Thank You for Applying to the Azure Services Invitations Program

* Indicates a required field.

1. Company name *
2. Business area *
3. If the answer for the above question is **other**, please specify
4. Role *
5. If the answer for the above question is **other**, please specify
6. Which service in the Azure Services Platform are you primarily interested in *
7. How did you hear about the Azure Services Platform *
8. If the answer for the above question is **other**, please specify
9. Software development skill level
10. Timeframe for developing with services in the Azure Services Platform

Submit Cancel

© 2009 Microsoft Corporation. All rights reserved. Microsoft Connect Terms of Use | Trademarks | Privacy Statement

Your Connect Dashboard - Windows Internet Explorer
https://connect.microsoft.com/dashboard/

Welcome, Leshik Sign out | Your Profile | Help
Choose one of Your Connections

HOME YOUR DASHBOARD CONNECTION DIRECTORY Search GO

Your Connect Dashboard

Select a Connection Select an Action

Activity in All Connections Since Your Last Login

Customize

SQL Server Data Services Programs

- SQL Server Data Services | Accept your invitation now
- SQL Server Data Services Pre-Registration | Pending | End Your Participation

Your Recent Activity

Your Discussions

No discussions available to display.

Your Bugs

No items available to display.

Your Suggestions

No items available to display.

Your Watch List

You currently have no items on your watch list.

Recommendations

Productivity Hub

The Productivity Hub is a SharePoint Server 2007 portal that is a customizable learning community preloaded with Office Online content. It c... read more
Not interested. Show me a new connection.

CISG

Connected Information Security Group read more
Not interested. Show me a new connection.

Heard in the Discussions

How many websites can SBS2008 host? Email?

From Connection: Windows Small Business Server 2008
Is there a limit to the number of websites that can be hosted on an SBS2008 server? How about email? We have about 5 websites that we h... read more
Posted by RSEM

Add Your Name to the Windows 7 Beta Test

From Connection: Connect Web Site Improvement
I would like to participate in the beta testing program for Windows 7. read more
Posted by Tom Lawrence

© 2009 Microsoft Corporation. All rights reserved. Microsoft Connect Terms of Use | Trademarks | Privacy Statement

Microsoft

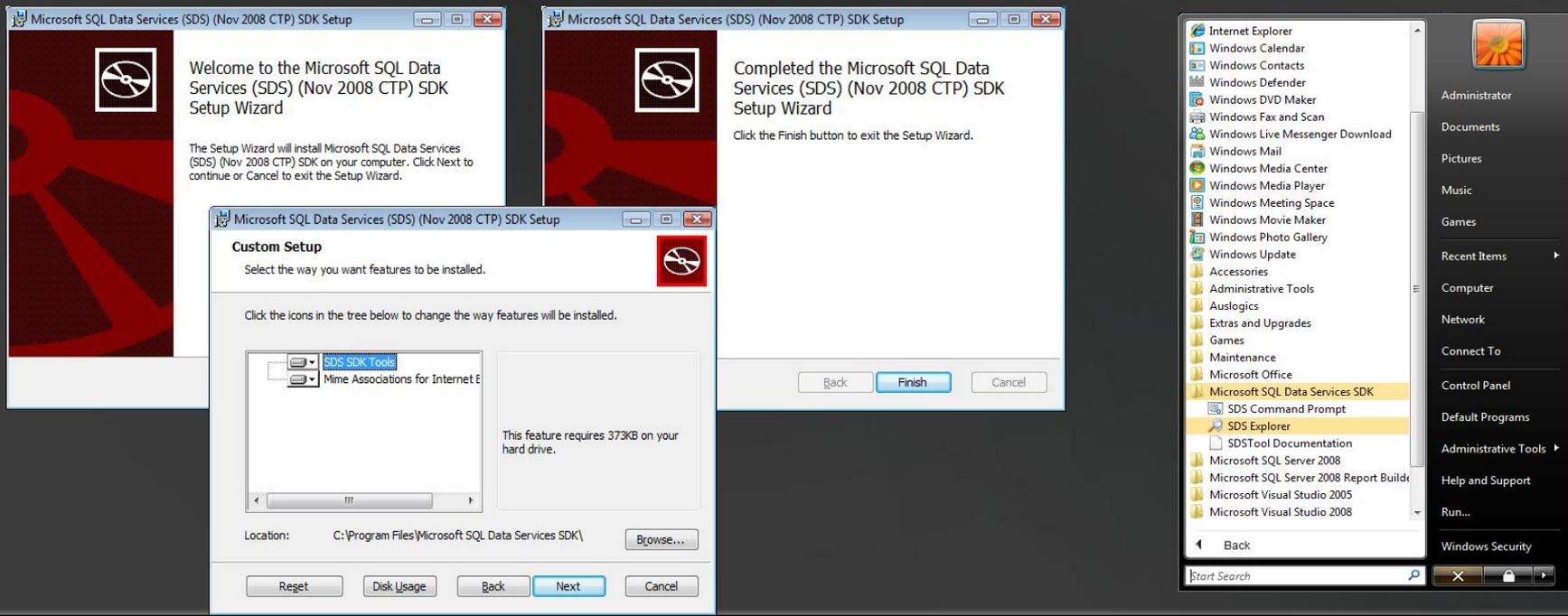
Создание облачного

The image shows a composite of three screenshots related to Azure development. On the left is the 'Azure Services Platform' sign-up page for .NET Services and SQL Services, which includes an 'Invitation Code' field and a 'Sign Up' button. In the center is the 'Azure Services Developer Portal' (TECH PREVIEW) showing a 'Project | Create a new service component' page with options for 'Storage Account', 'Hosted Services', and 'Live Services'. On the right is a 'New Project' dialog box in Visual Studio, where 'Blank Cloud Service' is selected as the project type and template. The dialog also shows the project name 'CloudService2', the location 'D:\My\Desktop', and the solution name 'CloudService2'.

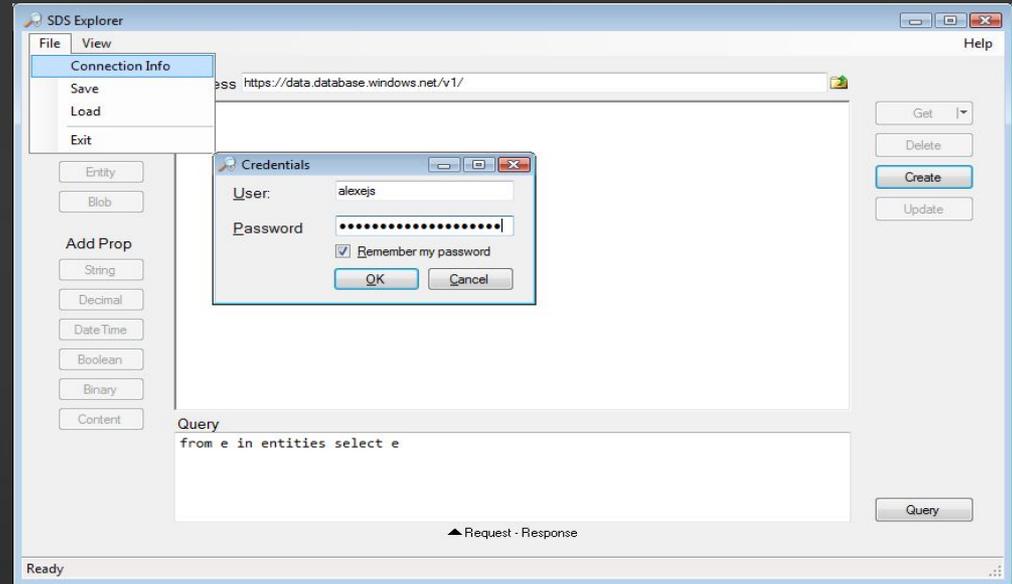
- Windows Azure SDK – примеры
- Windows Azure Tools for Microsoft Visual Studio – шаблоны проектов
- Подробнее – см., напр., <http://habrahabr.ru/blogs/Azure/49389/>

SDS SDK

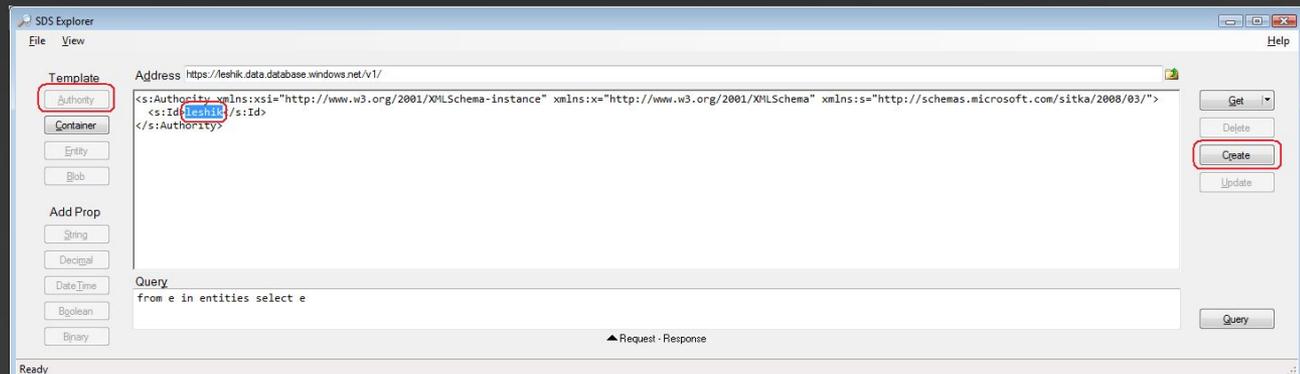
- Состав
 - SDS Explorer – графическая утилита для управления authorities / контейнерами / сущностями
 - Утилита командной строки SDS Command Prompt
- Размер – 553 КБ
- Ставится отсюда:
 - <https://www.microsoft.com/downloads/details.aspx?FamilyId=0B1FA5C6-EC9D-440B-939E-481DD05F2627&displaylang=en>



SDS Explorer

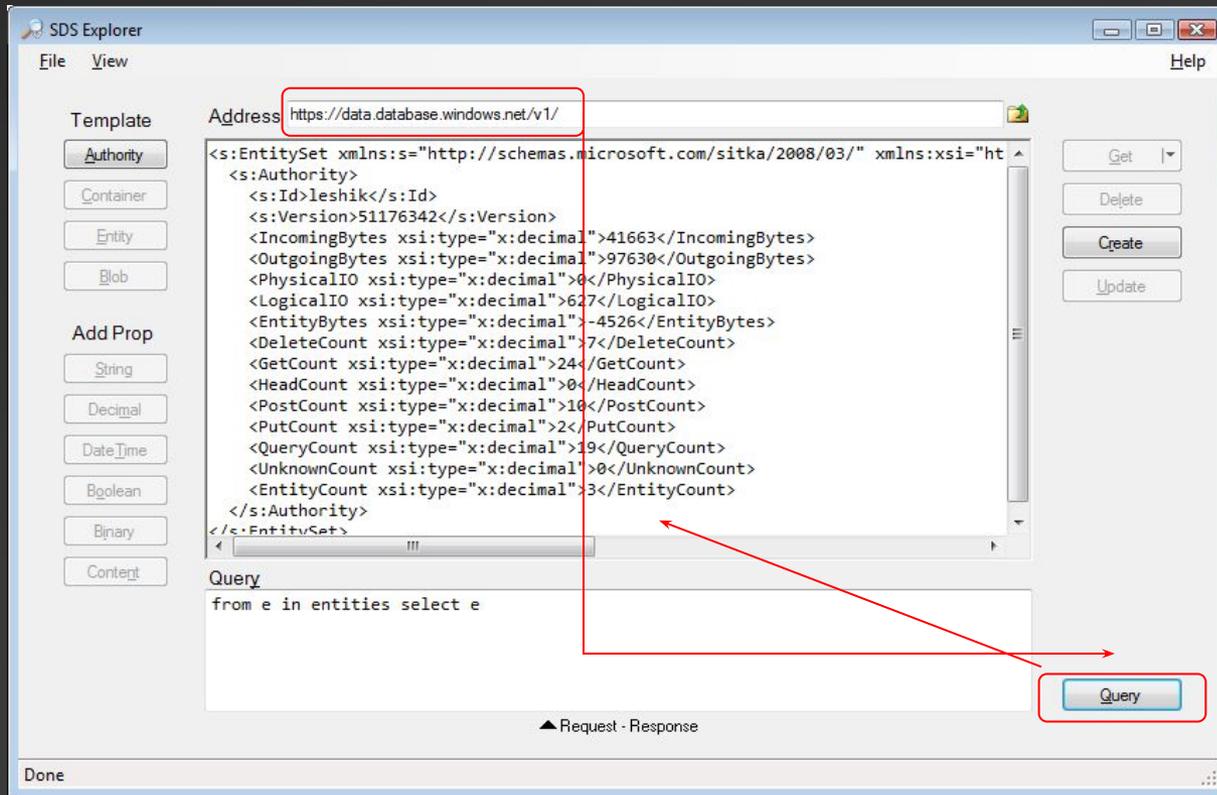


- Создание Authority



SDS Explorer

- Просмотр набора имеющихся у эккаунта authorities:



SDS Explorer

- Получение сущности

Metadata Properties (служебные поля)

Kind

Id

Version

The screenshot shows the SDS Explorer application window. The address bar contains the URL: `https://leshik.data.database.windows.net/v1/Библиотека/1`. The main content area displays XML metadata for an entity named 'Книга'. The XML is as follows:

```
<Книга xmlns:s="http://schemas.microsoft.com/sitka/2008/03/" xml:lang="ru">  
  <s:Id>1</s:Id>  
  <s:Version>51117358</s:Version>  
  <Название xsi:type="x:string">Медный всадник</Название>  
  <Жанр xsi:type="x:decimal">1</Жанр>  
</Книга>
```

Annotations in the image include:

- A red box around the address bar, with a red arrow pointing to the 'Kind' label.
- A red box around the 'Id' element in the XML, with a red arrow pointing to the 'Id' label.
- A red box around the 'Version' element in the XML, with a red arrow pointing to the 'Version' label.
- A red box around the 'Get' button in the right-hand panel, with a red arrow pointing to the 'Flexible Properties' label.

The 'Flexible Properties (пользовательские поля)' label is a blue box with white text. Below the XML, there is a 'Query' section with the following query:

```
from e in entities select e
```

At the bottom of the window, there is a 'Request - Response' section and a 'Done' status bar.

Гибкая схема сущности

```
<Книга xmlns:s="http://schemas.microsoft.com/sitka/2008/03/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:x="http://www.w3.org/2001/XMLSchema">
  <s:Id>1</s:Id>
  <Название xsi:type="x:string">Руслан и Людмила</Название>
  <Жанр xsi:type="x:string">ж1</Жанр>
</Книга>

<Книга xmlns:s="http://schemas.microsoft.com/sitka/2008/03/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:x="http://www.w3.org/2001/XMLSchema">
  <s:Id>2</s:Id>
  <Название xsi:type="x:string">Борис Годунов</Название>
  <Жанр xsi:type="x:string">ж1</Жанр>
  <Год_издания xsi:type="x:decimal">1831</Год_издания>
</Книга>

<Книга xmlns:s="http://schemas.microsoft.com/sitka/2008/03/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:x="http://www.w3.org/2001/XMLSchema">
  <s:Id>3</s:Id>
  <Название xsi:type="x:string">Герой нашего времени</Название>
  <Жанр xsi:type="x:string">ж1</Жанр>
  <Год_издания xsi:type="x:string">1841</Год_издания>
</Книга>

<Книга xmlns:s="http://schemas.microsoft.com/sitka/2008/03/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:x="http://www.w3.org/2001/XMLSchema">
  <s:Id>4</s:Id>
  <Название xsi:type="x:string">Введение в SDS</Название>
  <Жанр xsi:type="x:string">ж2</Жанр>
  <Кол-во_в_наличии xsi:type="x:decimal">10</Кол-во_в_наличии>
</Книга>
```

- Разные наборы свойств в разных сущностях
- Разные типы данных у свойств с одинаковым именем
- Каждая сущность создается отдельно

Запросы

- LINQ-подобный язык запросов

- Подробнее – см. BOL на SDS

(<http://msdn.microsoft.com/en-us/library/cc512417.aspx>)

- Пример

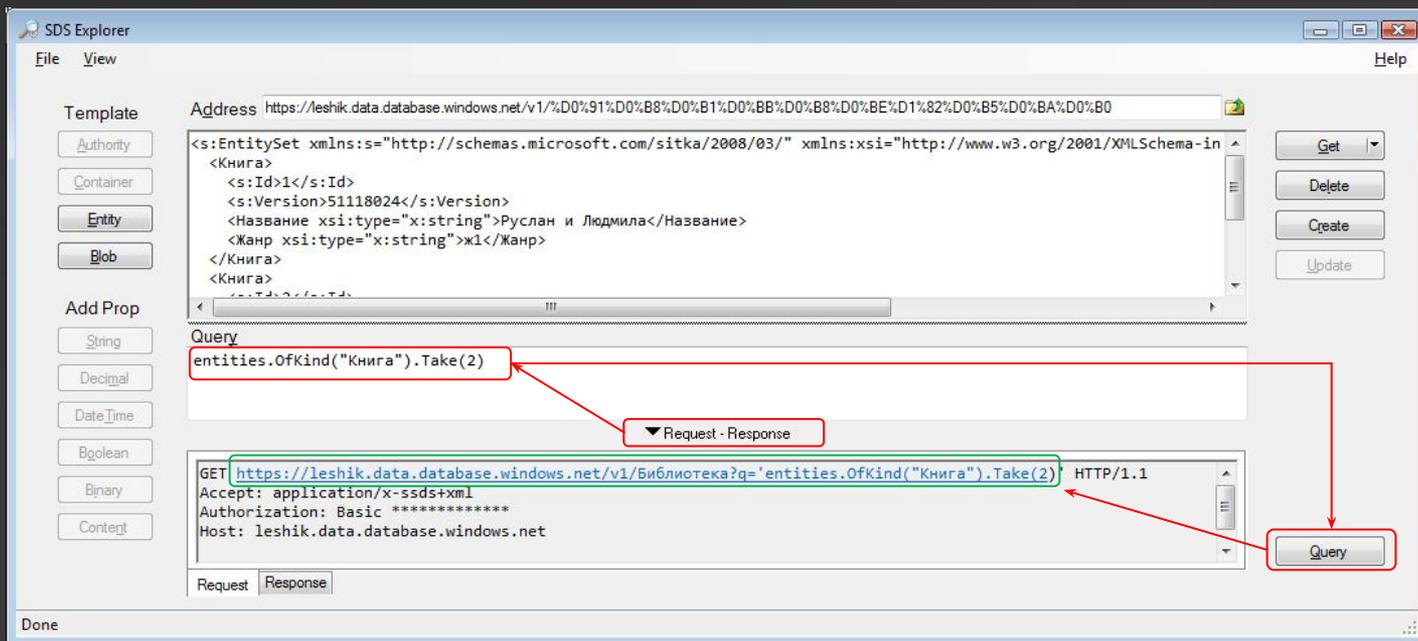
- (from жанр in entities.OfKind("Жанр")
where жанр["Жанр"] == "Классика"
from книга in entities.OfKind("Книга")
where книга["Жанр"] == жанр.Id
orderby книга["Название"]
select книга).Take(2)

- Комментарии

- Обращение к служебным св-вам: **книга.Id**, к пользовательским – **книга["Жанр"]**
- Case sensitive: `книга["жанр"]` не поймет
 - В то же время и ошибки не будет – схема-то гибкая
- `entities.OfKind("Жанр")` то же, что `where жанр.Kind == "Жанр": && ...`

Работа по протоколу REST

- HTTP-методы POST, GET, PUT, DELETE преобразуются в CRUD
 - Иными словами, в INSERT / SELECT / UPDATE / DELETE
- На самом деле мы ее все это время наблюдали
 - SDS Explorer – это приложение, работающее с SDS по REST
 - Имеется полезная кнопка Request-Response, которая позволяет посмотреть, что уходит/приходит по HTTP во время выполнения действий над объектами SDS
 - Существенно облегчает создание собственного REST-приложения



Создание собственного REST-приложения

- Все, что требуется – сформировать соответствующий HTTP Request и отправить его на URL SDS

```
Administrator: Command Prompt - powershell
C:\Users\alexejs>powershell
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

PS C:\Users\alexejs> c:\Events\09-02 TechDays\SDS\REST\SDS_REST_1.ps1

<s:EntitySet xmlns:s="http://schemas.microsoft.com/sitka/2008/03/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:x="http://www.w3.org/2001/XMLSchema-instance">
  <Книга>
    <s:Id>1</s:Id>
    <s:Version>51118024</s:Version>
    <Название xsi:type="x:string">Руслан и Людмила</Название>
    <Жанр xsi:type="x:string">ж1</Жанр>
  </Книга>
  <Книга>
    <s:Id>2</s:Id>
    <s:Version>51118025</s:Version>
    <Название xsi:type="x:string">Борис Годунов</Название>
    <Жанр xsi:type="x:string">ж1</Жанр>
    <Год_издания xsi:type="x:decimal">1831</Год_издания>
  </Книга>
</s:EntitySet>

PS C:\Users\alexejs>
```

```
. C:\Events\09-02 TechDays\SDS\REST\SDS_GetSitkaPwd.ps1
```

```
[string] $login = "alexejs"; [string] $password = GetSitkaPwd
```

```
function GetWebPage([string] $url)
```

```
{
  [net.httpWebRequest] $req = [net.webRequest]::create($url)
  $req.Credentials = new-object Net.NetworkCredential($login, $password)
  [net.httpWebResponse] $rep = $req.getResponse()
  [IO.StreamReader]$sr = new-object IO.StreamReader($rep.GetResponseStream())
  return $sr.ReadToEnd()
}
```

```
[string] $url = "https://leshik.data.database.windows.net/v1/Библиотека?q='entities.OfKind(`"Книга`")'.Take(2)'"
```

```
[string] $output = GetWebPage($url)
```

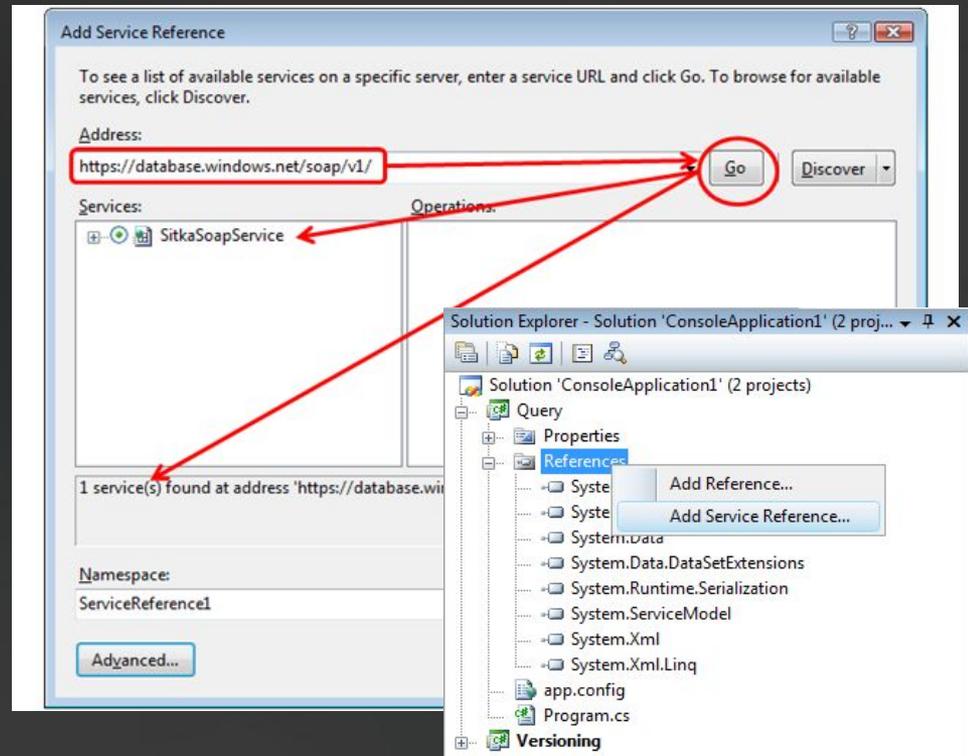
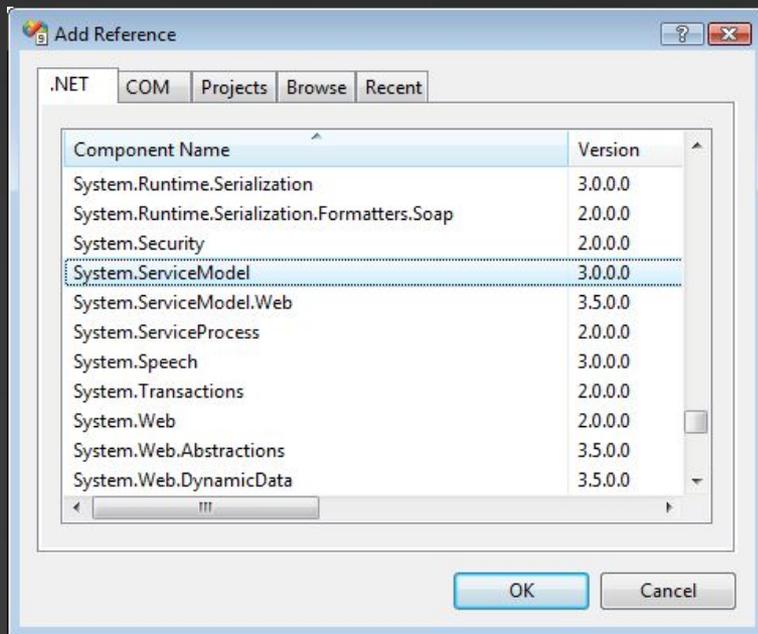
```
[console]::WriteLine(); [console]::WriteLine($output); [console]::WriteLine()
```

Взято из Request на пред.слайде



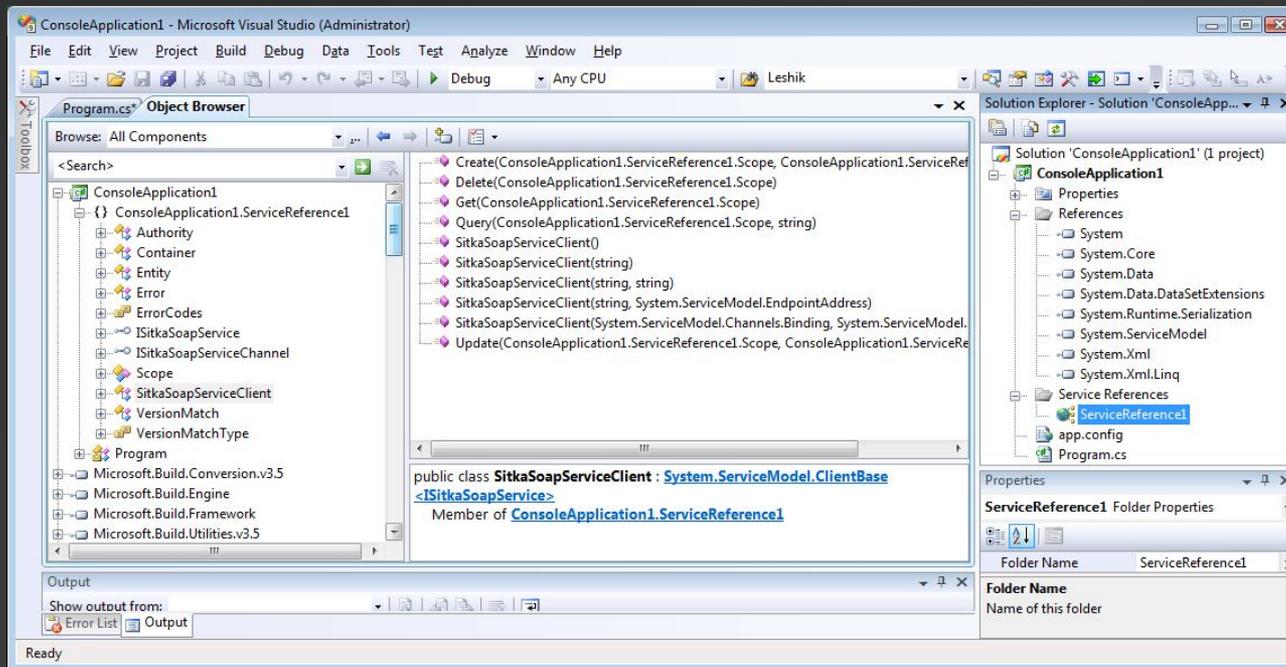
Работа по протоколу SOAP

- Создание ссылки на Windows Communication Foundation-сервис SDS
 - <https://database.windows.net/soap/v1/>



Работа по протоколу SOAP

- Читается wsdl, создается прокси
- Становится доступной объектная модель
 - IntelliSense



- Выполнение запроса
 - SitkaSoapServiceClient.Query(scope, «ТЕКСТ запроса»)

The screenshot shows the Visual Studio IDE with a C# program named Program.cs. The code defines a class Program with a static Main method. Inside Main, a SitkaSoapServiceClient proxy is created with the endpoint "BasicAuthEndpoint", username "alexajs", and password "Settings.Default.SitkaPassword". A Scope object is created with AuthorityId "leshik" and ContainerId "Библиотека". The proxy's Query method is called with the scope and the string "entities". A green arrow points from the word "запрос" to the "entities" string in the code.

```
using System;
using ConsoleApplication1.ServiceReference1;
using ConsoleApplication1.Properties;

class Program
{
    static void Main(string[] args)
    {
        SitkaSoapServiceClient proxy = new SitkaSoapServiceClient("BasicAuthEndpoint");
        proxy.ClientCredentials.UserName.UserName = "alexajs";
        proxy.ClientCredentials.UserName.Password = Settings.Default.SitkaPassword;

        Scope scope = new Scope();
        scope.AuthorityId = "leshik";
        scope.ContainerId = "Библиотека";

        Entity[] e = proxy.Query(scope, "entities");
    }
}
```

The Locals window shows the following variables:

Name	Value	Type
proxy	{ConsoleApplication1.ServiceReference1.SitkaSoapServiceClient}	ConsoleApplication1.ServiceReference1.SitkaSoapServiceClient
scope	{ConsoleApplication1.ServiceReference1.Scope}	ConsoleApplication1.ServiceReference1.Scope
e	{ConsoleApplication1.ServiceReference1.Entity[6]}	ConsoleApplication1.ServiceReference1.Entity[]
e[0]	{ConsoleApplication1.ServiceReference1.Entity}	ConsoleApplication1.ServiceReference1.Entity
e[1]	{ConsoleApplication1.ServiceReference1.Entity}	ConsoleApplication1.ServiceReference1.Entity
e[2]	{ConsoleApplication1.ServiceReference1.Entity}	ConsoleApplication1.ServiceReference1.Entity
e[3]	{ConsoleApplication1.ServiceReference1.Entity}	ConsoleApplication1.ServiceReference1.Entity
ExtensionData	{System.Runtime.Serialization.ExtensionDataObject}	System.Runtime.Serialization.ExtensionDataObject
extensionDataField	{System.Runtime.Serialization.ExtensionDataObject}	System.Runtime.Serialization.ExtensionDataObject
Id	"4"	string
IdField	"4"	string
Kind	"Книга"	string
KindField	"Книга"	string
Properties	Count = 3	System.Collections.Generic.Dictionary<string,object>
Properties[0]	{[Название, Введение в SDS]}	System.Collections.Generic.KeyValuePair<string,object>
Properties[1]	{[Жанр, ж2]}	System.Collections.Generic.KeyValuePair<string,object>
Properties[2]	{[Кол-во_наличии, 10]}	System.Collections.Generic.KeyValuePair<string,object>
Raw View		
PropertiesField	Count = 3	System.Collections.Generic.Dictionary<string,object>
PropertyChanged	null	System.ComponentModel.PropertyChangedEventHandler
Version	57903981	long
VersionField	57903981	long
e[4]	{ConsoleApplication1.ServiceReference1.Entity}	ConsoleApplication1.ServiceReference1.Entity
e[5]	{ConsoleApplication1.ServiceReference1.Entity}	ConsoleApplication1.ServiceReference1.Entity

Аутентификация по SOAP

- В отличие от REST, помимо базовой, поддерживаются:
 - `// Access Control Service`
 - `proxy = new SitkaSoapServiceClient("UsernameTokenEndpoint");`
 - `proxy.ClientCredentials.UserName.UserName = "solutionname";`
 - `proxy.ClientCredentials.UserName.Password = "solutionpassword";`
 - `//CardSpace`
 - `proxy = new SitkaSoapServiceClient("CardSpaceTokenEndpoint");`
 - `proxy.DisplayInitializationUI();` //вызывает CardSpace prompt
 - `proxy.Open();` //для CardSpace требуется явное открытие прокси
 - `//X.509`
 - `proxy = new SitkaSoapServiceClient("CertificateTokenEndpoint");`
 - `proxy.ClientCredentials.ClientCertificate.SetCertificate(`
 - `"CN=localhost", StoreLocation.LocalMachine, StoreName.My`
 - `);` //предварительно требуется создать сертификат (с закрытым ключом) и установить его куда-нибудь на данную машину (например, в Local Computer\Personal\Certificates), Personal Container – это StoreName.My).
 - //Также создать самогенерирующийся открытый ключ сертификата в Local Computer\Trusted People\Certificates, чтобы он валидировался
 - Адрес службы <https://data.database.windows.net/soap/v1/zurich>

```
static void Main(string[] args)
{
    SitkaSoapServiceClient proxy = new SitkaSoapServiceClient("BasicAuthEndpoint");
    proxy.ClientCredentials.UserName.UserName = "alexejs";
    proxy.ClientCredentials.UserName.Password = Settings.Default.SitkaPassword;

    Scope scope = new Scope();
    scope.AuthorityId = "leshik"; scope.ContainerId = "Библиотека"; scope.EntityId = "1";

    Entity entity = proxy.Get(scope);
    Debug.WriteLine("Старое значение версии = " + entity.Version);

    entity.Properties["Название"] = "Руслан и Людмила";

    scope.VersionMatch = new VersionMatch();
    scope.VersionMatch.Version = entity.Version;
    scope.VersionMatch.MatchType = VersionMatchType.Match;

    try
    { scope = proxy.Update(scope, entity); }

    catch (System.ServiceModel.FaultException<Error> e)
    {
        string errMsg;
        switch (e.Detail.StatusCode)
        {
            case ErrorCodes.EntityNotFound: errMsg = String.Format("Эвона! А сущность с Id {0} уже  
кто-то прихлопнул :(", entity.Id); break;
            case ErrorCodes.PreconditionFailed: errMsg = String.Format("Дык ить сущность с Id {0} с  
тех пор вроде как того, поменялась :(", entity.Id); break;
            default: errMsg = "Чета случилось, а че - ктож ево знать :("; break;
        }
        Debug.WriteLine(errMsg);
    }
    Debug.WriteLine("Новое значение версии = " + scope.VersionMatch.Version);
    proxy.Close();
}
```

Совместны
й доступ

То же самое в случае REST (1)

```
static void Main(string[] args)
{
    string login = "alexejs", password = Settings.Default.SitkaPassword;
    string authorityId = "leshik", containerId = "Библиотека", entityId = "1";
    string url = "https://" + authorityId + ".data.database.windows.net/v1/" + containerId + "/" +
entityId;
    XmlDocument entityXml = new XmlDocument();

    HttpWebRequest req = (HttpWebRequest) WebRequest.Create(url);
    req.Credentials = new NetworkCredential(login, password);
    req.ContentType = "application/x-ssds+xml";

    //Читаем сущность с Id = entityId в entityXml (XmlDocument).
    req.Method = "GET";
    HttpWebResponse rep = (HttpWebResponse) req.GetResponse();
    string xml_сущности_в_виде_текста = new StreamReader(rep.GetResponseStream()).ReadToEnd();
    entityXml.LoadXml(xml_сущности_в_виде_текста);
    rep.Close();
    Debug.WriteLine("\nУспешно прочитана сущность:\n" + xml_сущности_в_виде_текста);

    //Из XML вытаскиваем и запоминаем версию
    XmlNamespaceManager nsmgr = new XmlNamespaceManager(entityXml.NameTable);
    nsmgr.AddNamespace("s", "http://schemas.microsoft.com/sitka/2008/03/");
    XmlNode x = entityXml.DocumentElement.SelectSingleNode("s:Version", nsmgr);
    string версия = x.InnerText;

    //Меняем какие-нибудь св-ва сущности
    entityXml.SelectSingleNode("Книга/Название").InnerText = "Руслан и Людмила";
    xml_сущности_в_виде_текста = entityXml.InnerXml;
    Debug.WriteLine("Изменяем сущность к виду:\n" + xml_сущности_в_виде_текста);
    ...
}
```

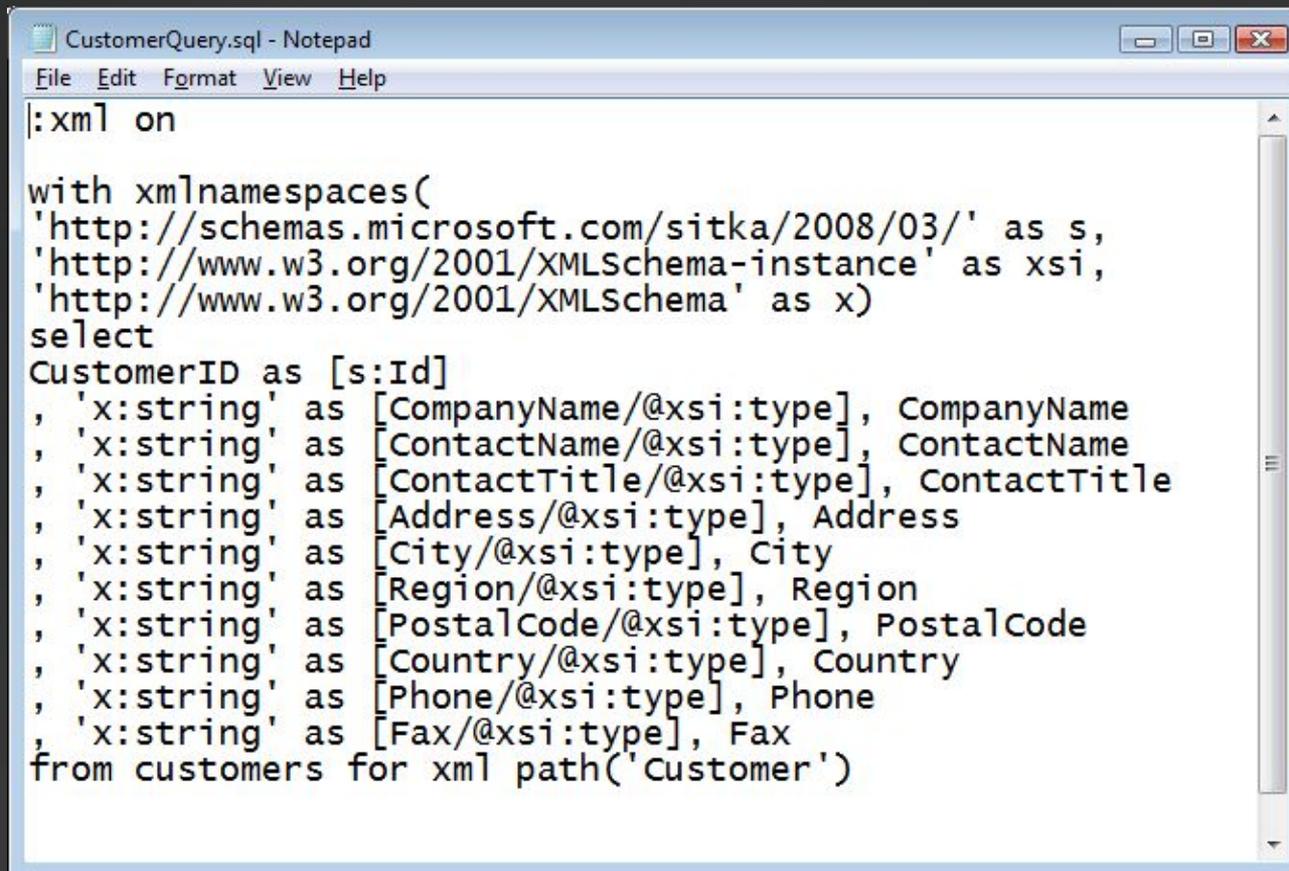
То же самое в случае REST (2)

```
//Апдейтим сущность на сервер
req = (HttpWebRequest)WebRequest.Create(url);
req.Credentials = new NetworkCredential(login, password);
req.ContentType = "application/x-ssds+xml";
req.Method = "PUT";
req.Headers["If-Match"] = версия; //если на сервере будет уже другая версия, выкинется исключение
Encoding кодировка = Encoding.UTF8;
req.ContentLength = кодировка.GetByteCount(xml_сущности_в_виде_текста);
req.GetRequestStream().Write(кодировка.GetBytes(xml_сущности_в_виде_текста), 0, (int)req.ContentLength);
try
{
    rep = (HttpWebResponse)req.GetResponse();
    Debug.WriteLine("Обновление прошло успешно. Новая версия записи = " + rep.Headers["eTag"]);
    rep.Close();
}
catch (WebException e)
{
    rep = (HttpWebResponse)e.Response;
    if (e.Status == WebExceptionStatus.ProtocolError)
        switch (rep.StatusCode)
        {
            case HttpStatusCode.NotFound:
                Debug.WriteLine(String.Format("Эвона! А сущность с Id {0} уже кто-то прихлопнул :(",
entityId)); break;
                //rep.StatusDescription == "Not Found"
                //e.Message == "The remote server returned an error: (404) Not Found."
            case HttpStatusCode.PreconditionFailed:
                Debug.WriteLine(String.Format("Дык ить сущность с Id {0} с тех пор вроде как того,
поменялась :(", entityId)); break;
                //rep.StatusDescription == "A precondition, such as Version, could not be met."
                //e.Message == {"The remote server returned an error: (412) A precondition, such as Version,
could not be met."
                }
            else
                Debug.WriteLine("Чета случилось, а че - ктож яво таперича знать :(");
        }
    Debug.WriteLine("");
}
}
```

Массовая загрузка

- Подготовить из таблицы XML с определениями сущностей
 - Превращаем каждую запись в сущность
 - Делается элементарным запросом SELECT ... FOR XML
- Загрузить полученный XML с определениями сущностей на SDS
 - Делается при помощи утилиты st.exe (вторая в составе SDK)
 - Мне неизвестно о существовании у SDS специализированных интерфейсов массовой загрузки
 - Поэтому я полагаю, что чуда не происходит
 - Утилита разбивает пакетный XML на куски по числу заказанных потоков
 - Чтобы на каждый поток пришлось примерно равное кол-во сущностей
 - И создает их по заказанному протоколу аналогично рассмотренным выше способам

Массовая загрузка

A screenshot of a Notepad window titled "CustomerQuery.sql - Notepad". The window contains a SQL query designed to output XML. The query starts with "with xmlnamespaces" and lists three namespaces: "http://schemas.microsoft.com/sitka/2008/03/" as 's', "http://www.w3.org/2001/XMLSchema-instance" as 'xsi', and "http://www.w3.org/2001/XMLSchema" as 'x'. The "select" clause includes "CustomerID as [s:Id]" and a list of columns: "CompanyName", "ContactName", "ContactTitle", "Address", "City", "Region", "PostalCode", "Country", "Phone", and "Fax". Each column is wrapped in an "x:string" data type and mapped to an XML element name (e.g., [CompanyName/@xsi:type]). The query concludes with "from customers for xml path('Customer')".

```
CustomerQuery.sql - Notepad
File Edit Format View Help
|:xml on

with xmlnamespaces(
'http://schemas.microsoft.com/sitka/2008/03/' as s,
'http://www.w3.org/2001/XMLSchema-instance' as xsi,
'http://www.w3.org/2001/XMLSchema' as x)
select
CustomerID as [s:Id]
, 'x:string' as [CompanyName/@xsi:type], CompanyName
, 'x:string' as [ContactName/@xsi:type], ContactName
, 'x:string' as [ContactTitle/@xsi:type], ContactTitle
, 'x:string' as [Address/@xsi:type], Address
, 'x:string' as [City/@xsi:type], City
, 'x:string' as [Region/@xsi:type], Region
, 'x:string' as [PostalCode/@xsi:type], PostalCode
, 'x:string' as [Country/@xsi:type], Country
, 'x:string' as [Phone/@xsi:type], Phone
, 'x:string' as [Fax/@xsi:type], Fax
from customers for xml path('Customer')
```

- `sqlcmd -S (local) -d Northwind -E -i c:\temp\CustomerQuery.sql -o Customers.xml`

Массовая загрузка

```
C:\Windows\system32\cmd.exe

C:\Temp>"C:\Program Files"\Microsoft SQL Data Services SDK\st.exe loadbatch leshik Northwind c:\Temp\Customers.xml /rest /overwrite /verbose /parallel:4
* Options: /rest /host:data.database.windows.net /user:alexejs /password:*****
***** /parallel:4 /verbose
* Upsert entity batch into 'https://leshik.data.database.windows.net/v1/Northwind'
...
* Created entity 'ANATR'
* Created entity 'ALFKI'
* Created entity 'ANTON'
* Created entity 'AROUT'
* Created entity 'BERGS'
* Created entity 'BLONP'
* Created entity 'BLAUS'
* Created entity 'BOLID'
* Created entity 'BONAP'
* Created entity 'BOTTM'
* Created entity 'BSBEU'
* Created entity 'CACTU'
* Created entity 'CENIC'
* Created entity 'CHOPS'
* Created entity 'COMMI'
* Created entity 'CONSH'
* Created entity 'DRACD'
* Created entity 'DUMON'
```

```
C:\Windows\system32\cmd.exe

* Created entity 'SUPRD'
* Created entity 'THEBI'
* Created entity 'THECR'
* Created entity 'TOMSP'
* Created entity 'TORTU'
* Created entity 'TRADH'
* Created entity 'TRAIH'
* Created entity 'UINET'
* Created entity 'UAPFE'
* Created entity 'UICTE'
* Created entity 'WANDK'
* Created entity 'WARTH'
* Created entity 'WELLI'
* Created entity 'WHITC'
* Created entity 'WILMK'
* Created entity 'WOLZA'
* Processed 91 entries using 4 workers
* Completed in 37552 milliseconds
* Throughput 2.423266 op/s
Success

C:\Temp>pause
Press any key to continue . . . _
```

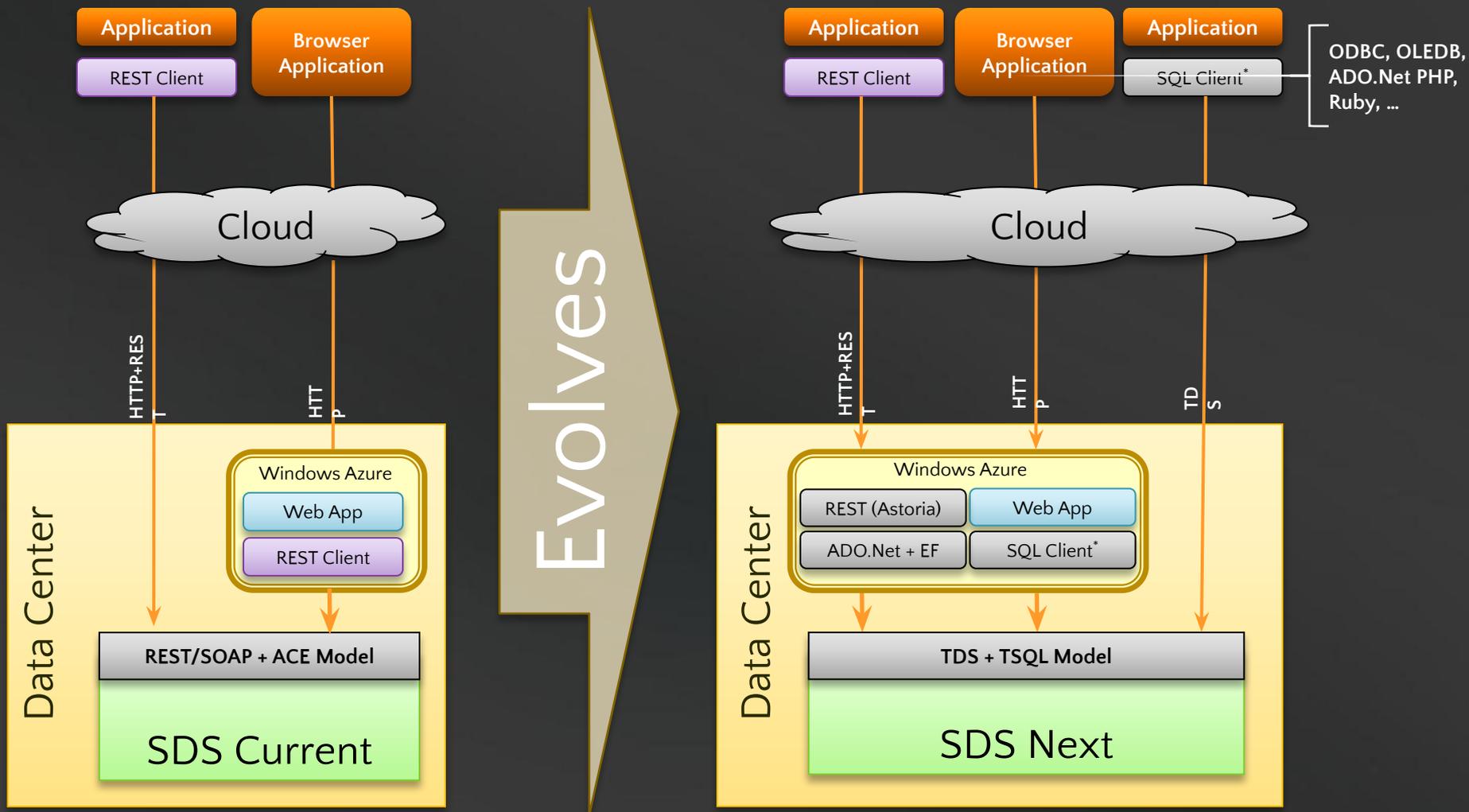
- "%programfiles%\Microsoft SQL Data Services SDK\st.exe loadbatch leshik Northwind c:\Temp\Customers.xml /rest /overwrite /verbose /parallel:4

Чтоб ты жил в эпоху перемен

(старинное китайское проклятие)

- На конференции MIX'09 в марте было объявлено (<http://msdn.microsoft.com/en-us/sqlserver/dataservices/default.aspx>), что SDS будут поддерживать произвольный T-SQLный доступ по обычному протоколу TDS
- Таким образом, работа с SQL Server в Облаке не будет отличаться от работы с локальным SQL Server
 - Это здорово, но означает, что только что рассказанное уже не актуально
- До мая закрытое тестирование, май – июнь – по приглашениям, с июля – все желающие
- Модель ACE и доступ по REST/SOAP в настоящее время остаются, но впоследствии перейдут в Azure Storage

То же самое на картинке



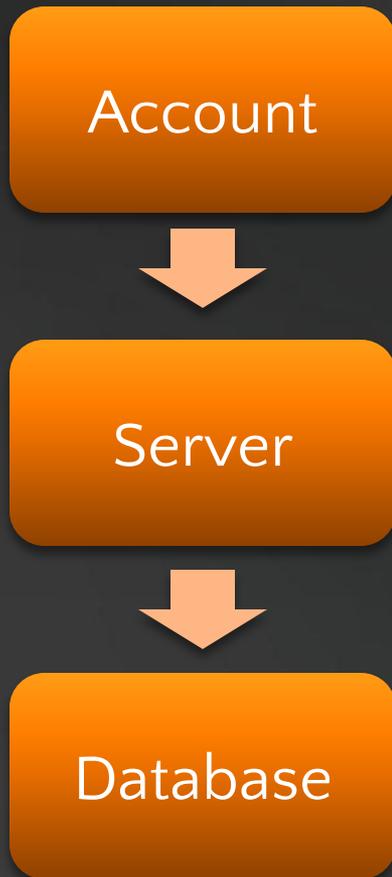
* Client access enabled using TDS for ODBC, ADO.Net, OLEDB, PHP-SQL, Ruby, ...

Преимущества новой модели

Просто БД в Облаке

- Сочетание преимуществ привычной работы с SQL Server и неограниченной доступности и масштабируемости Облака ценой минимальных административных издержек
 - Знакомая реляционная модель SQL Server
 - Использование существующих инструментов и API
 - Поддерживается работа из PHP, Ruby, Java

Service Provisioning Model

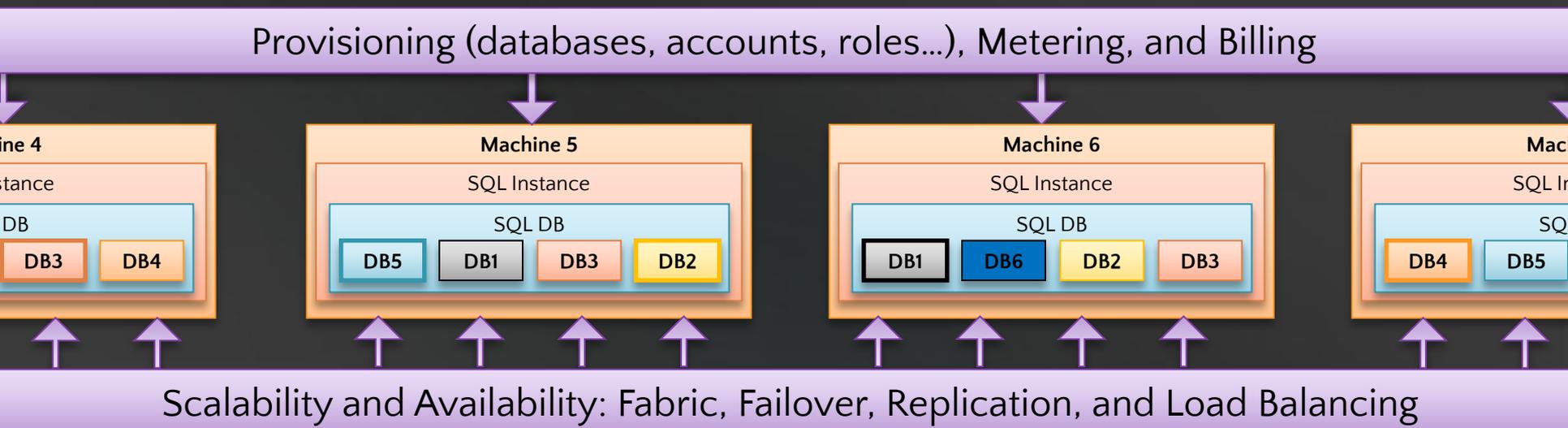


- Каждый *account*
 - объект биллинга
 - владеет одним или несколькими *виртуальными серверами*
- Каждый *server*
 - хостит одну или несколько БД, включая *virtual master*
 - каждая БД ограничена в размерах
 - На сервере заводятся *logins*
- Каждая *database*
 - имеет много *SQL users*

Server=*server1*.data.database.windows.net
Database=*testDB*
Login=*nigele*[@*server1*] (отображается на *testuser*)

Логические базы данных (ЛБД)

- По-прежнему автоматическая репликация, избыточность, контроль здоровья и восстановление в случае сбоя



Пример в рамках новой модели

```
<?php
$host = "server.data.dev.msds.com";
$dbname = "database";
$dbuser = "user@server";
$dbpwd = "password";
$driver = "{SQL Server Native Client 10.0}";

// Build connection string
$dsn="Driver=$driver;Server=$host;Database=$dbname;Encrypt=true;TrustServerCertificate=true";
if (!($conn = @odbc_connect($dsn, $dbuser, $dbpwd)) ) {
    die("Connection error: " . odbc_errormsg());
}

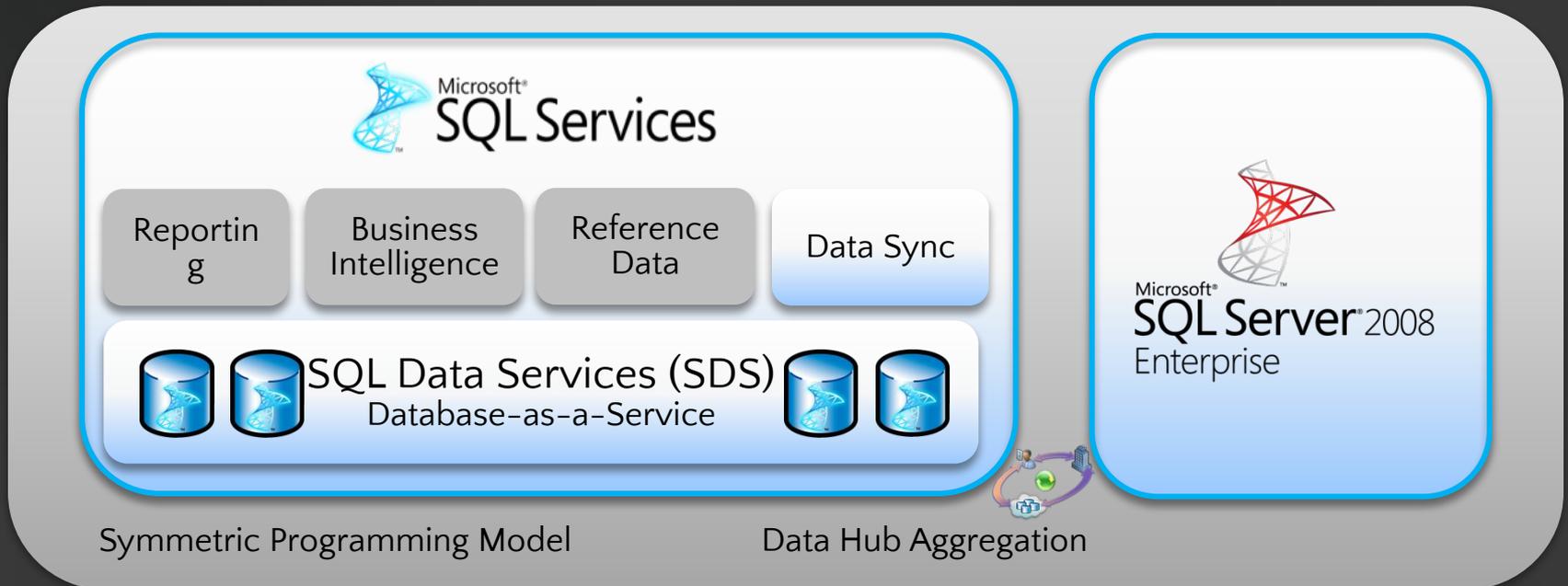
// Got a connection, run simple query
if ($qh = @odbc_exec($conn, "SELECT A, B FROM myTable")) {
    // Dump query result
    $rows = 0;
    while ( $row = @odbc_fetch_object($qh) ) {
        echo("$rows: $row->A $row->B\r\n");
        $rows++;
    }

    @odbc_free_result($qh);
}
else {
    // Error running query
    echo("Query error: " . odbc_errormsg($conn));
}

// Free the connection
@odbc_close($conn);
?>
```

- <http://blogs.msdn.com/ssds/archive/2009/03/27/9515262.aspx>

Расширяя SQL в Облако



- Предложение начального уровня – базовая функциональность СУБД в качестве сервиса (SDS), синхронизация и концентрация данных
- Дальнейшие предложения
 - Дополнительные сервисы платформы данных: BI/DSS, DW
 - Новые сервисы: Reference Data, Secure Data Hub

Вопросы?

Microsoft[®]

Your potential. Our passion.[™]

© 2007 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.
MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.