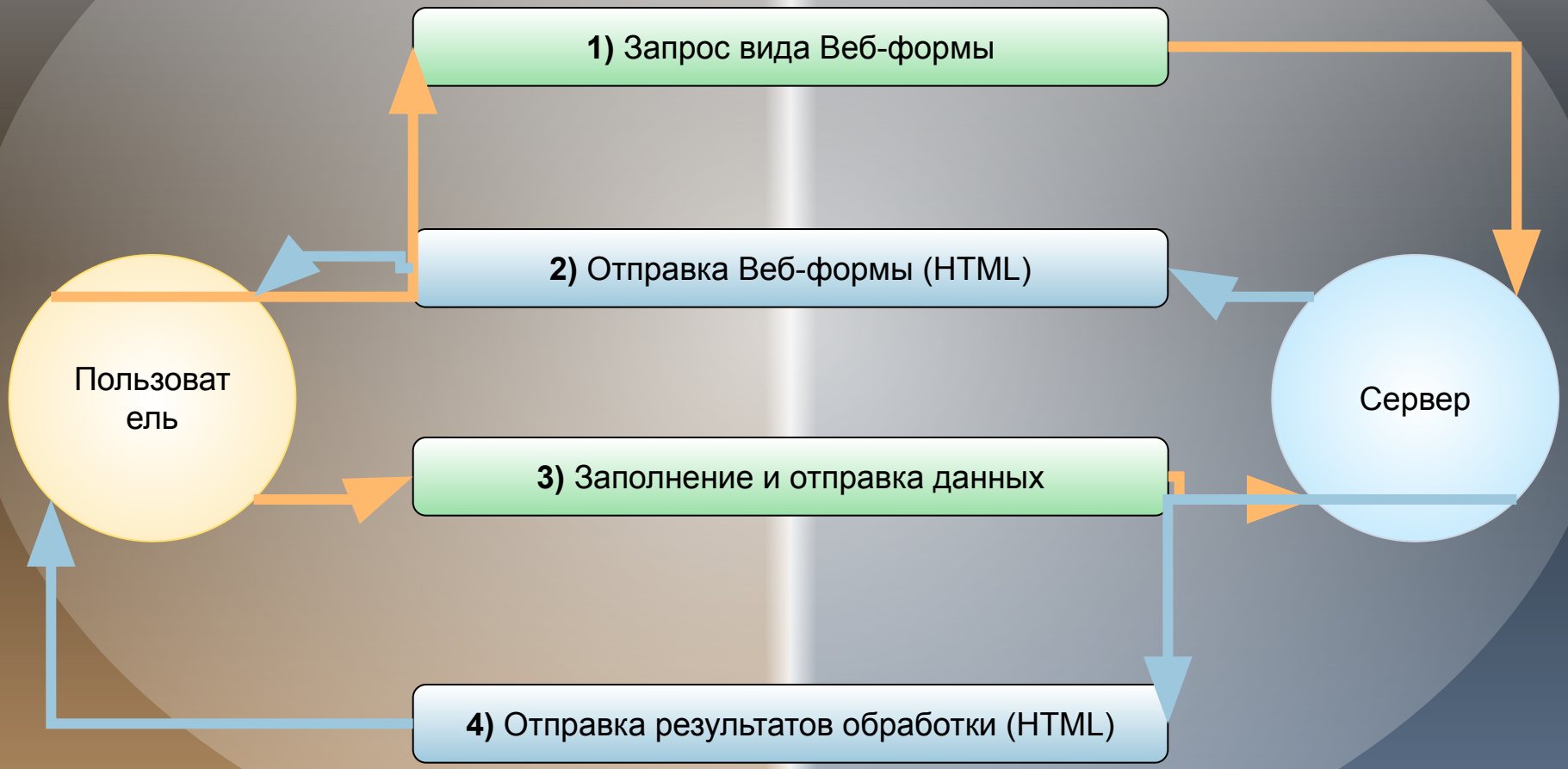
Применение

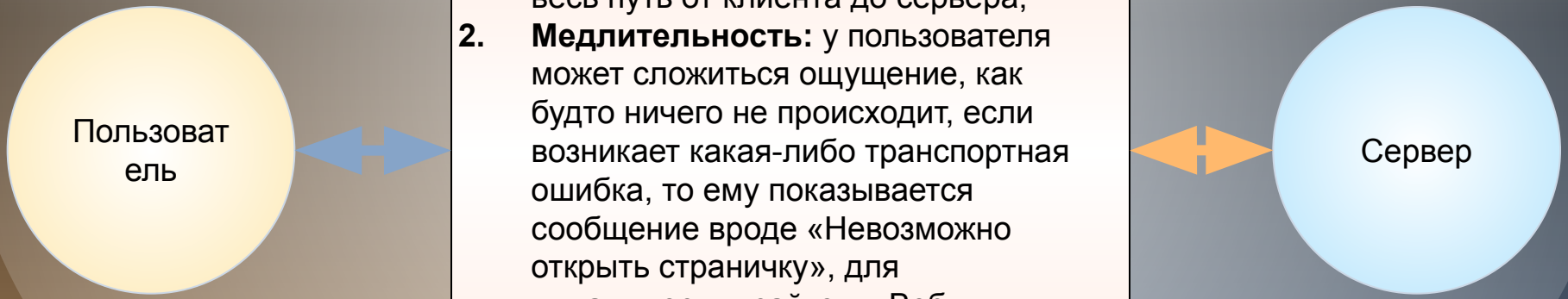
Microsoft ASP.NET AJAX

Андрей Скляревский
.NET Developer
andrew@oridea.org

Обычная модель динамического Веб-сайта

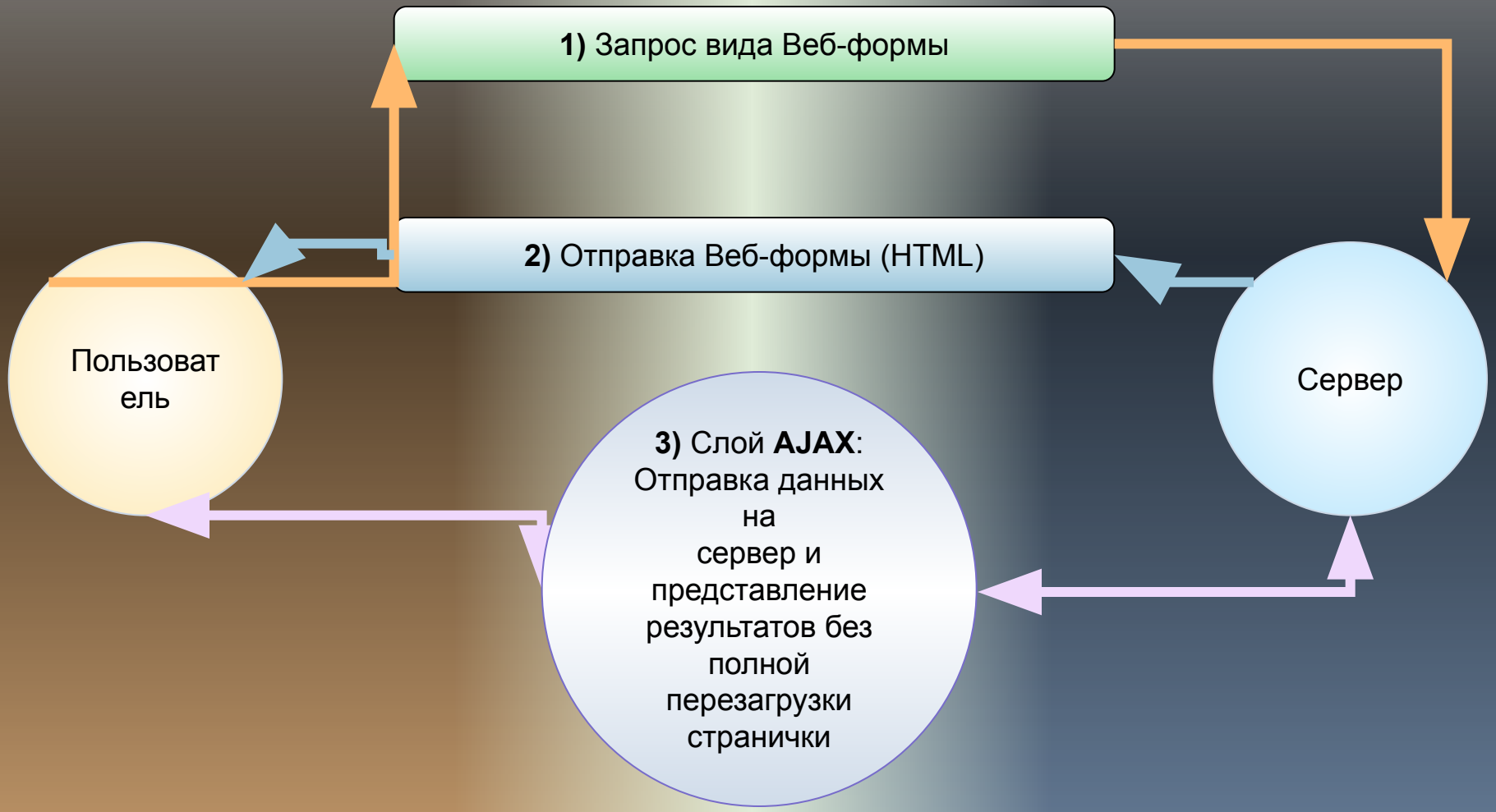


Проблемы обычной модели Веб-сайта



1. **Неэффективность:** много лишних запросов – один и тот же код (HTML, JavaScript, etc.) много раз проходит весь путь от клиента до сервера;
2. **Медлительность:** у пользователя может сложиться ощущение, как будто ничего не происходит, если возникает какая-либо транспортная ошибка, то ему показывается сообщение вроде «Невозможно открыть страничку», для динамических сайтов и Веб-приложений, часто бывает, что команда «Обновить» не решает проблемы.

Модель Веб-сайта с использованием **AJAX** (Asynchronous JavaScript And XML)



Обновление Веб-странички при использовании AJAX

1. Загружается базовый HTML с сервера;
2. Пользователь вводит данные или срабатывает заданный таймер обновления;
3. При помощи JavaScript отправляется запрос на сервер, содержащий необходимую информацию о действиях пользователя;
4. Сервер отправляет необходимые данные клиенту, которые могут представлять собой HTML-код, простой текст или сериализованные объекты;
5. На клиенте данные преобразуются и отображаются (HTML-код или текст вставляется в соответствующий элемент, если от сервера пришли сериализованные объекты, они десериализуются и на их основе генерируются деревья элементов DOM);
6. Пользователь может вновь модифицировать данные или добавить новые, которые затем снова могут быть отправлены на сервер для сохранения. Таким образом, полной перезагрузки странички не происходит.

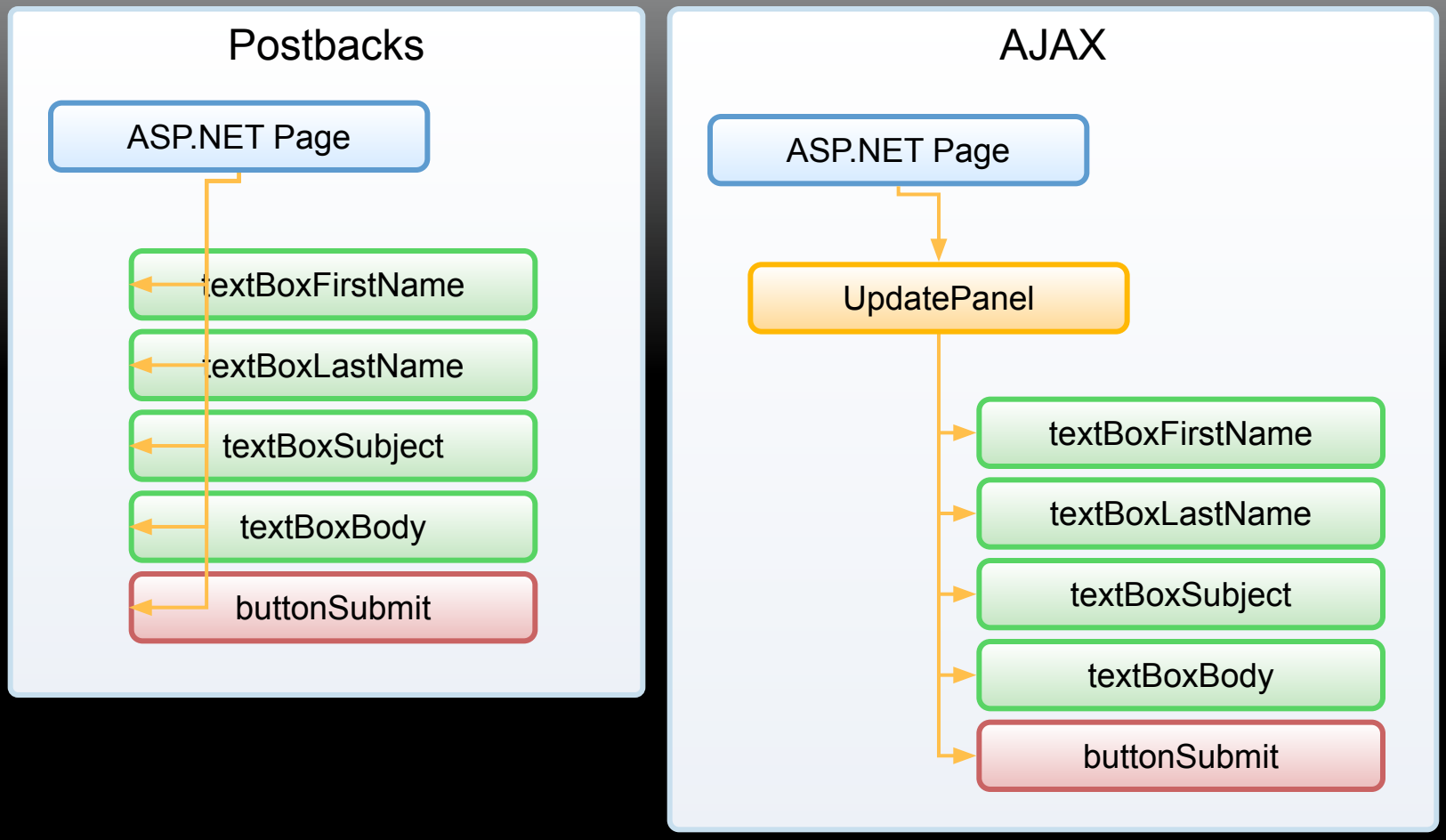
Компоненты ASP.NET AJAX

1. ASP.NET 2.0 AJAX Extensions – главный компонент ASP.NET AJAX, являющийся набором функциональности, предназначенной для исполнения на серверной стороне. Позволяет проводить Веб-разработку с использованием AJAX практически точно так же, как и традиционную разработку Веб-сайтов ASP.NET.
2. Microsoft AJAX Library – набор JavaScript скриптов, объединённых в общую библиотеку. Позволяет проводить взаимодействие серверного кода с клиентским, скрывая технические детали, не меняющиеся от проекта к проекту. Этот набор доступен для скачивания с Веб-сайта ASP.NET AJAX отдельно от AJAX Extensions разработчикам на других платформах.
3. ASP.NET AJAX Control Toolkit – дополняет функциональность ASP.NET AJAX Extensions, содержит элементы управления, которые разработчик может использовать как они есть (as-is) или дополнив их нужными функциями.

Основные методы внедрения технологии AJAX на странички ASP.NET

- UpdatePanel: Новый элемент управления, представляющий собой особеннную панель, которая позволяет обновлять выбранный регион странички без её полной перезагрузки. При этом можно использовать элементы управления из стандартного набора ASP.NET, а так же и практически любые другие – обычно достаточно просто поместить их внутрь UpdatePanel, чтобы они начали работать без полной перезагрузки странички (postbacks);
- Вызов серверных Веб-сервисов из клиентского JavaScript кода: ASP.NET AJAX Extensions может сгенерировать JavaScript-обёртку для обычного XML Веб-сервиса, что позволяет использовать методы этого Веб-сервиса прямо в исполняющимся в браузере JavaScript коде, модифицируя затем содержание странички самостоятельно через DOM-модель;

ASP.NET AJAX Extensions: UpdatePanel

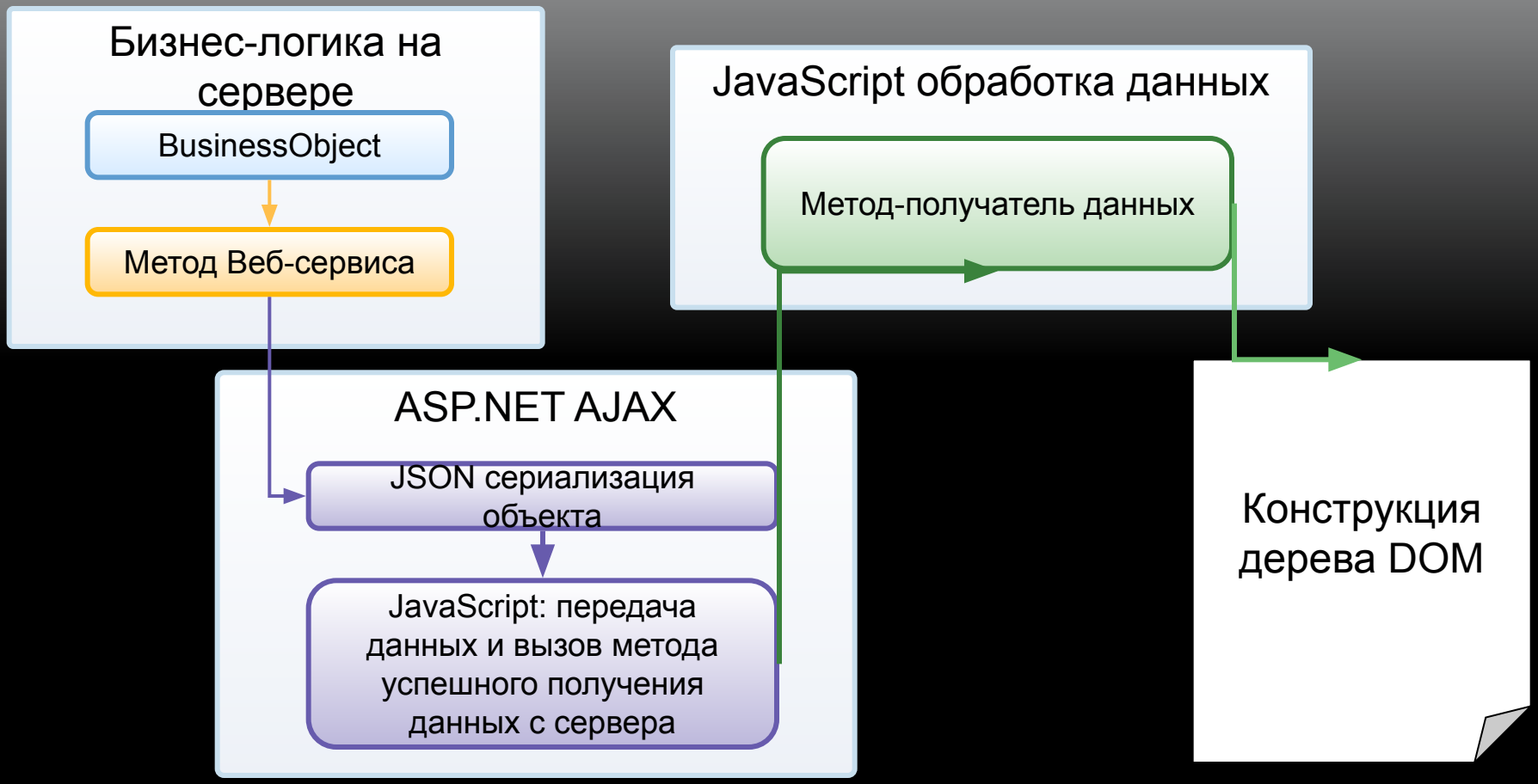


- Для внедрения технологии AJAX иногда достаточно лишь вставить UpdatePanel на уже существующую страничку. Кроме этого, обычно необходимо ещё немного модифицировать работу «валидаторов» и некоторых других элементов управления.

ASP.NET AJAX Extensions: UpdatePanel

DEMO

ASP.NET AJAX Extensions: Вызов метода Веб-сервиса на клиенте



- ASP.NET AJAX позволяет легко создать Веб-сервис, методы которого можно будет использовать не только по протоколу SOAP, но и на стороне клиента, используя JavaScript.

ASP.NET AJAX Extensions: Вызов метода Веб-сервиса на клиенте

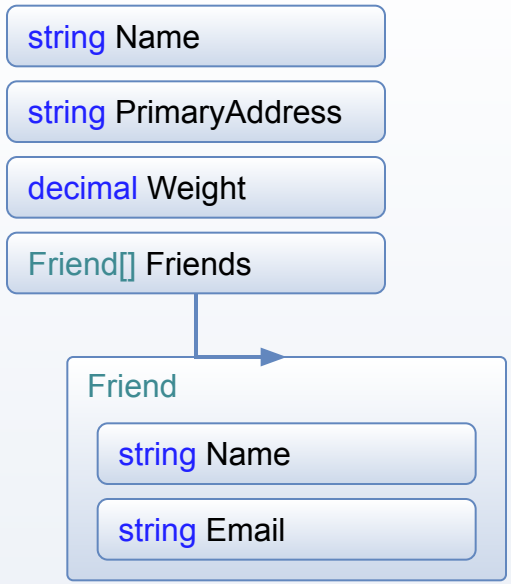
DEMO

Что такое JSON?

JSON

```
var contact = {
     “Name”: “John Smith",
     “PrimaryAddress": “john@smith.com”,
     “Weight”: 80,
     “Friends": [
       {
            “Name": “Alex",
            “Email": “alex@example.net"
       },
       {
            “Name": “Frank",
            “Email": “frank@example.net"
       }
     ]
};
```

Объект «contact»



- JavaScript Object Notation – хорошая альтернатива XML, когда речь идёт о быстродействии и удобстве процессов сериализации и десериализации объектов на клиенте и сервере;
- ASP.NET AJAX Extensions содержит классы для сериализации и десериализации объектов с использованием JSON, который стандартно используется для объектов передаваемых в Веб-сервис и им возвращаемых, при этом существует возможность модифицировать алгоритм сериализации.

Модификация алгоритма JSON-сериализации

Алгоритм JSON-сериализации можно модифицировать, создав класс, наследующийся от *System.Web.Script.Serialization.JavaScriptConverter* и реализующий методы *Deserialize*, *Serialize* и свойство *SupportedTypes*.

JavaScriptConverter предназначен только для изменения сериализуемых данных, но не способа генерации JSON. Метод *Serialize* принимает объект для сериализации, а возвращает *IDictionary<string, object>*, которая уже и преобразовывается в JSON. Метод *Deserialize*, соответственно, принимает *IDictionary<string, object>* и тип объекта *(System.Type),* а возвращает объект.

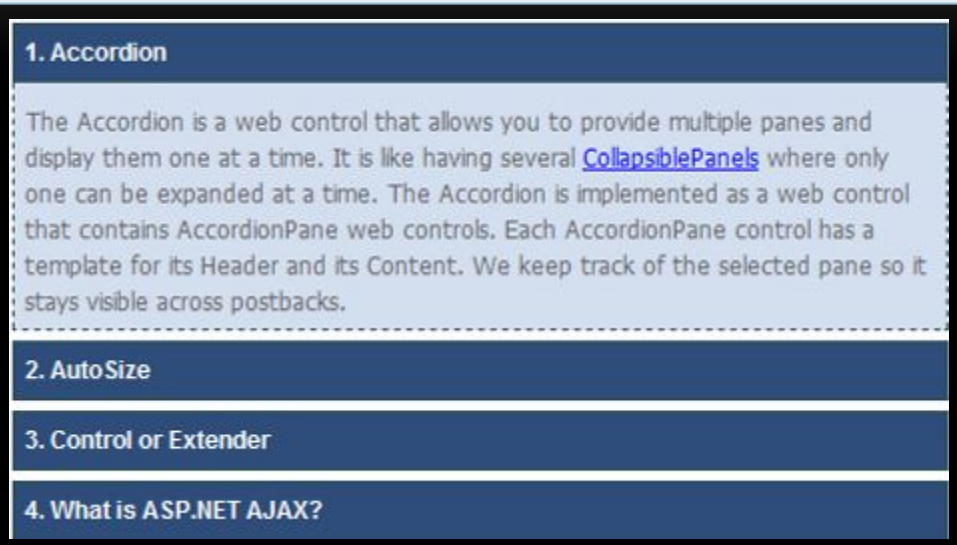
Свою реализацию *JavaScriptConverter* нужно так же зарегистрировать, либо используя метод *JavaScriptSerializer.RegisterConverters*, либо, более удобным в большинстве случаев способом – модификацией Web.config:

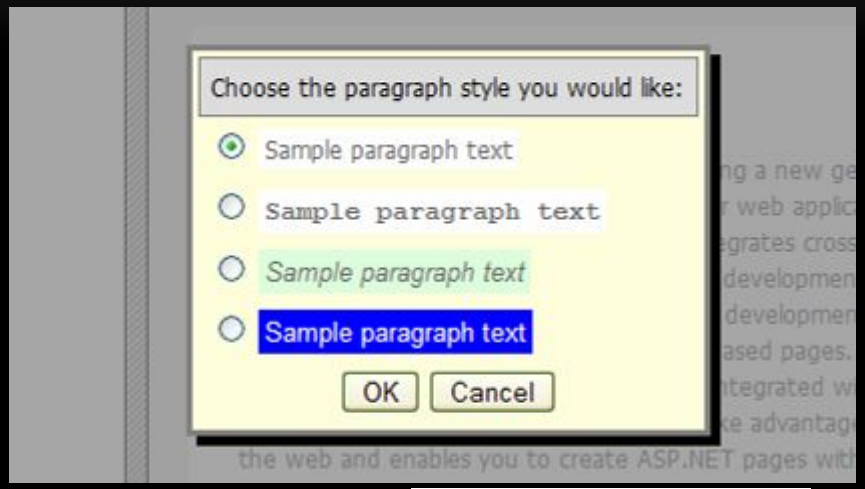
```
<jsonSerialization>
  <converters>
    <add name="MyJsonConverter" type="Project.MyJsonConverter"/>
  </converters>
</jsonSerialization>
```

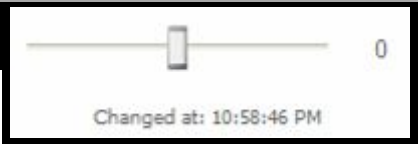
В секции system.web.config / scripting / webServices.

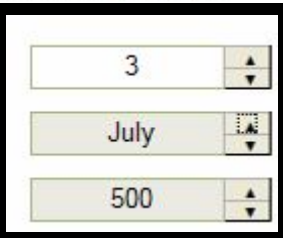
ASP.NET AJAX Control Toolkit

- Содержит большое количество универсальных элементов управления, подобные которым ранее приходилось заново разрабатывать в разных проектах;
- Протестирован в нескольких браузерах и поддерживается Microsoft, как и другие компоненты ASP.NET AJAX;
- Элементы управления можно использовать прямо как они есть;









Вопросы?