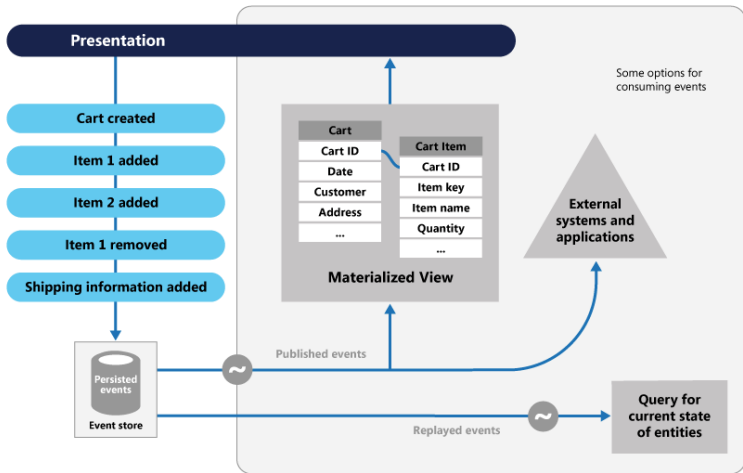


Talk started

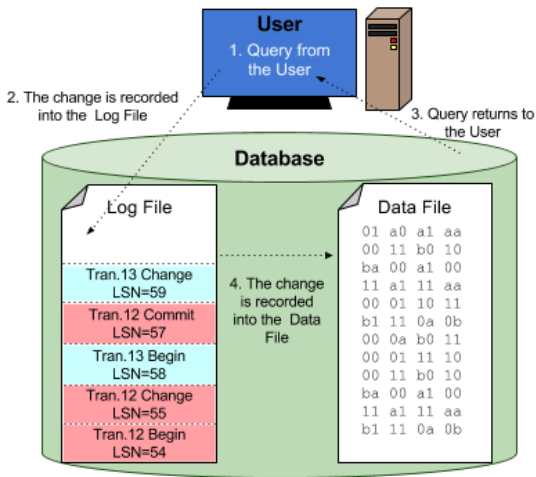
Talk ended

New tool added to the toolbox

**What the hell Event Sourcing (ES) is and
what that changes in my life?**



	123 id	123 featureMetadataId	ABC featureConfigurationId	createDate	updateDate
1	3	1	c5226c85-7106-fc00-2de6-c183eadfb163	2020-08-28 07:22:02	2020-08-28 07:22:02
2	4	1	c5226c85-7106-fc00-2de6-c183eadfb163	2020-08-28 07:24:51	2020-08-28 07:24:51



source

legend

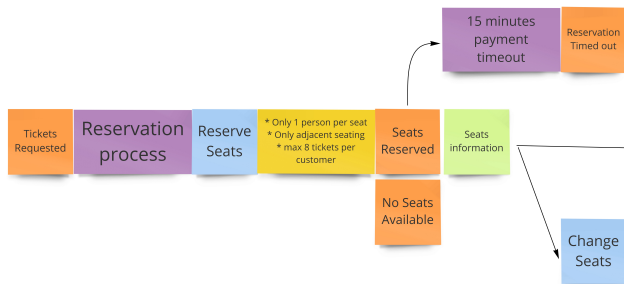
Domain
Events

Actions

Information

Consistent
Business Rules

Eventual
Consistent
Business Rules



source

**You can't change the
Past!**

- ▶ **Event**
- ▶ **Event Store**
- ▶ **Event Stream**
- ▶ **Projection**
- ▶ **Snapshot**

**Let's take a look into the
code**

”Versioning in an Event Sourced System”

Gregory Young

1337

SPE4K

The neat trick from a
Michigan Mom that 80%
of developers don't know
Just click!



The art of destroying software

Lesson learned

- ▶ **Weak schema**
- ▶ **Event versioning**
- ▶ **Saving all stack**
- ▶ **Snapshots**
- ▶ **Avoid "and"**
- ▶ **Errors**
 - ▶ compensation action
 - ▶ deleting a stream is ok
 - ▶ copy and replace
- ▶ **public/private streams**
- ▶ **two aggregates one stream**
- ▶ **one aggregate two streams**
- ▶ **copy and transform → migrations**

Weak schema

```
1      {  
2          "id": 1,  
3          "firstName": "John"  
4          "lastName" : "Smith" ,  
5          "age" : 25  
6      }
```

```
1      public class Client  
2      {  
3          public int Id { get; set; }  
4          public string FirstName { get; set; }  
5          public string LastName { get; set; }  
6          public int Age { get; set; }  
7      }
```

Weak schema

newtonsoft.com/json/help/html/DeserializeObject.htm

mentation

cumentation

JSON

an Object

a Collection

a Dictionary

JSON to a file

with JsonConverters

a DataSet

Raw JSON value

Unindented JSON

Conditional Property

ize an Object

ize a Collection

ize a Dictionary

ize an Anonymous Type

ize a DataSet

ize with

CreationConverter

ize JSON from a file

an Object

torHandling setting

reationHandling setting

alueHandling setting

ememberHandling setting

eHandling setting

ceLoopHandling setting

ReferencesHandling

matHandling setting

eZoneHandling setting

neHandling setting

aPropertyHandling setting



Deserialize an Object

This sample deserializes JSON to an object.

Sample

Types

```
public class Account
{
    public string Email { get; set; }
    public bool Active { get; set; }
    public DateTime CreatedDate { get; set; }
    public IList<string> Roles { get; set; }
}
```

Usage

```
string json = @"{
    'Email': 'james@example.com',
    'Active': true,
    'CreatedDate': '2013-01-20T00:00:00Z',
    'Roles': [
        'User',
        'Admin'
    ]
}";
```

```
Account account = JsonConvert.DeserializeObject<Account>(json);
```

```
Console.WriteLine(account.Email);
// james@example.com
```

Mapping

```
1         Map.To<Client>()  
2             .WhenAbsent("age", -99)  
3             .WhenAbsentError("id")
```

Wrapping

```
1     public class ClientWrapper {  
2         private readonly JObject _json;  
3  
4         public ClientWrapper(string json) {  
5             _json = JObject.Parse(json);  
6         }  
7  
8         public Guid Id {  
9             get { return Guid.Parse(_json["id"]); }  
10            set { _json["id"] = value.ToString(); }  
11        }  
12        (...)  
13    }
```

Event versioning

- ▶ Double publish (v1, v2)
- ▶ Constructor with previous version

Example

```
1      ClientCreated_v2 ConvertFrom(ClientCreated_v1 e) {  
2          return new ClientCreated_v2(e.Id, -99);  
3      }  
4  
5      ClientCreated_v1 ConvertFrom(ClientCreated_v2 e) {  
6          return new ClientCreated_v1(e.Id);  
7      }
```


Sabing all stack

```
1  public void Apply(ItemSold e) {  
2      _tax = e.Subtotal * .22;  
3  }
```

Sabing all stack

```
1  public void Apply(ItemSold e) {  
2      if(e.Date < Convert.ToDateTime("1/1/2017")) {  
3          _tax = e.Subtotal * .22;  
4      } else {  
5          _tax = e.Subtotal * .23;  
6      }  
7  }
```

Sabing all stack

```
1 public void SellItem (...) {  
2     //snip  
3     var tax = subtotal * .23;  
4     Apply(new ItemSold(_id ,.... , tax));  
5 }  
6  
7 public void SellItem (...) {  
8     //snip  
9     var customerScore = _externalService  
10         .GetCustomerScoreFor(customer);  
11  
12     Apply(new ItemSold(_id ,.... , customerScore));  
13 }
```

Lesson learned

- ▶ **Weak schema ✓**
- ▶ **Event versioning ✓**
- ▶ **Saving all stack ✓**
- ▶ **Snapshots**
- ▶ **Avoid "and"**
- ▶ **Errors**
 - ▶ compensation action
 - ▶ deleting a stream is ok
 - ▶ copy and replace
- ▶ **public/private streams**
- ▶ **two aggregates one stream**
- ▶ **one aggregate two streams**
- ▶ **copy and transform → migrations**