Saritasa Tools

Или ещё один подход к архитектуре приложения

Иван Кожин, Очень Интересно

Наш зоопарк проектов

Случай на проекте

Заказчик: У меня при заказе товара происходит ошибка...

Случай на проекте

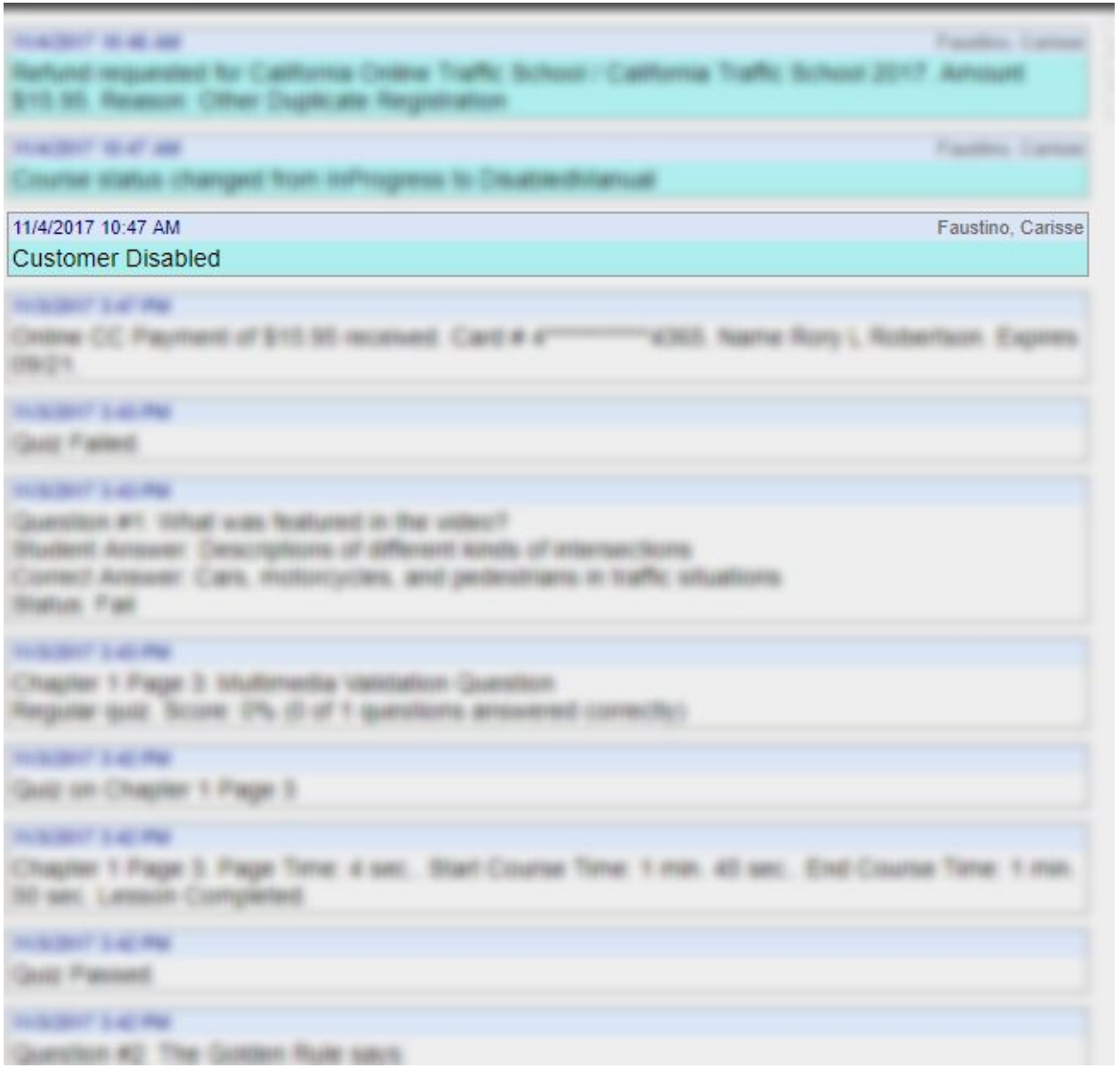
```
System.NullReferenceException...
```

```
14:06:23 [Application Exception] System.Web.HttpUnhandledException (0x80004005): Exception of type 'System.Web.HttpUnhandledException' was thr
> an instance of an object.
   at System.Collections.Generic.Dictionary`2.Insert(TKey key, TValue value, Boolean add)
   at System.Collections.Generic.Dictionary`2.Add(TKey key, TValue value)
   at Tube.AddPageToTube(String requestId) in W:\crm\Source\Utilities\HelpersSite\WebService\tube.cs:line 61
   at Tube.AddPageToTube() in W:\crm\Source\Utilities\HelpersSite\WebService\tube.cs:line 51
   at SaberMasterPage.get_TubeId() in W:\crm\Source\CRMSite\_masters\SaberMasterPage.master.cs:line 60
   at ASP._masters_sabermasterpage_master.__Render__control1(HtmlTextWriter __w, Control parameterContainer) in w:\crm\Source\CRMSite\_masters
   at System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection children)
   at System.Web.UI.Control.RenderChildren(HtmlTextWriter writer)
   at System.Web.UI.Control.Render(HtmlTextWriter writer)
   at System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter adapter)
   at System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter)
   at System.Web.UI.Control.RenderControl(HtmlTextWriter writer)
   at System.Web.UI.Control.RenderChildrenInternal(HtmlTextWriter writer, ICollection children)
   at System.Web.UI.Control.RenderChildren(HtmlTextWriter writer)
   at System.Web.UI.Page.Render(HtmlTextWriter writer)
   at System.Web.UI.Control.RenderControlInternal(HtmlTextWriter writer, ControlAdapter adapter)
   at System.Web.UI.Control.RenderControl(HtmlTextWriter writer, ControlAdapter adapter)
   at System.Web.UI.Control.RenderControl(HtmlTextWriter writer)
   at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
   at System.Web.UI.Page.HandleError(Exception e)
   at System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
   at System.Web.UI.Page.ProcessRequest(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)
   at System.Web.UI.Page.ProcessRequest()
```

Случай на проекте

Мы: Ок, а какой аккаунт вы использовали?
Заказчик: test@someproject.com
Мы: На какой странице и какой продукт вы запрашивали?
...

Случай на проекте #2



Ну, извините…

Цели

- Разработать гибкую типовую архитектуру для наших проектов.

Цели

- Разработать гибкую типовую архитектуру для наших проектов.
- Заложить в архитектуру возможность журналирования событий в проекте.

Цели

- Разработать гибкую типовую архитектуру для наших проектов.
- Заложить в архитектуру возможность журналирования событий в проекте.
- Предоставить возможность анализа состояния системы в различное время.

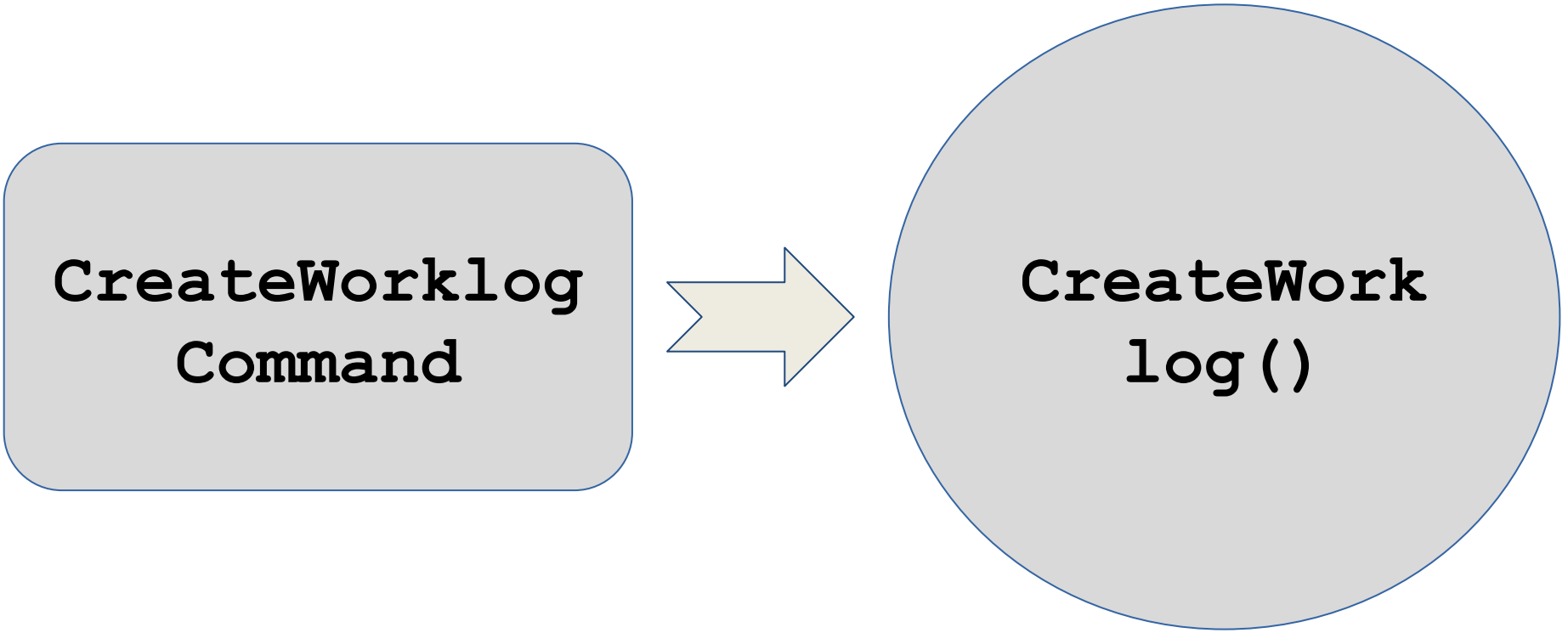
Цели

- Разработать гибкую типовую архитектуру для наших проектов.
- Заложить в архитектуру возможность журналирования событий в проекте.
- Предоставить возможность анализа состояния системы в различное время.
- Улучшить возможности интеграционного тестирования наших проектов.

Паттерн «Команда»

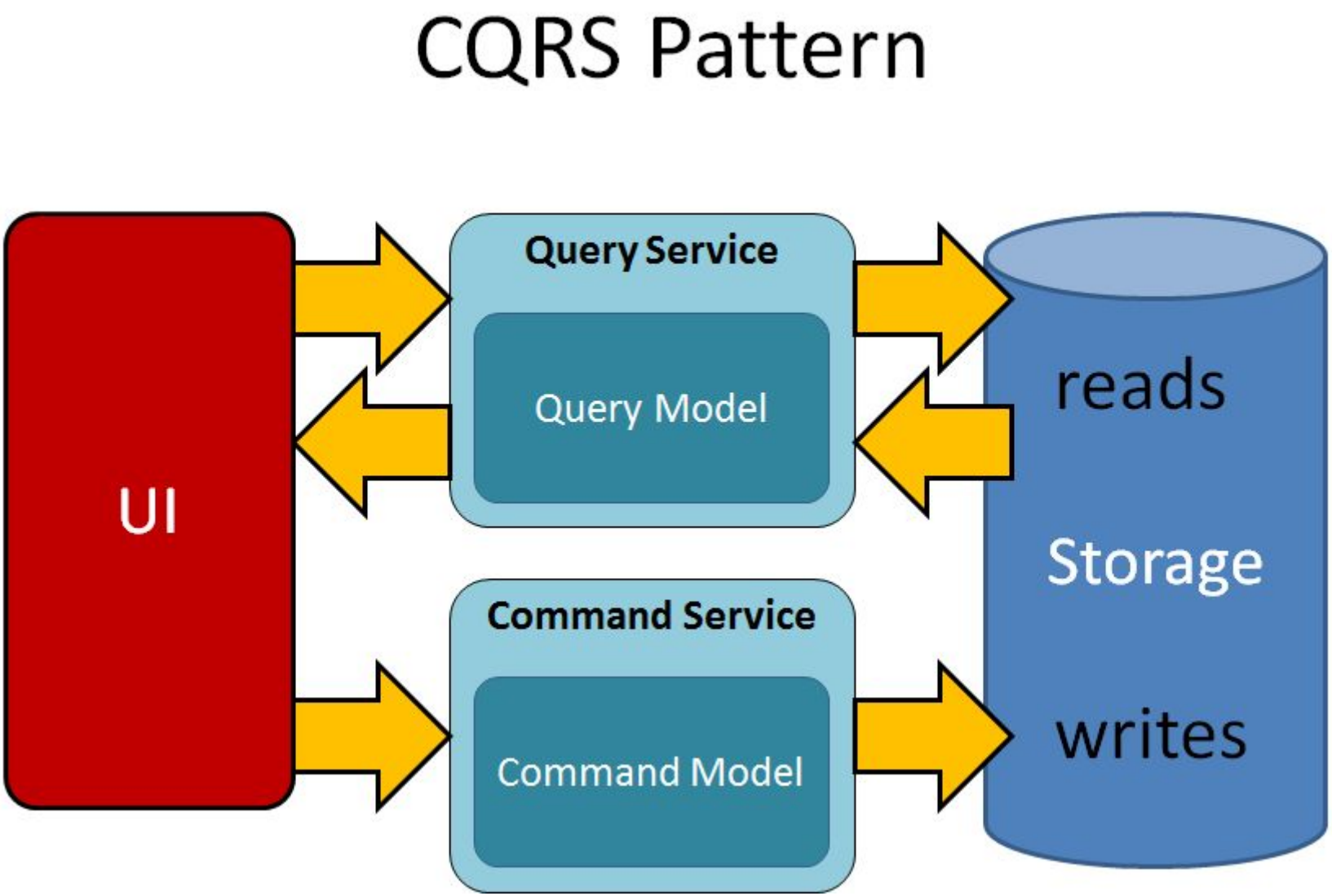
Команда

Обработчик

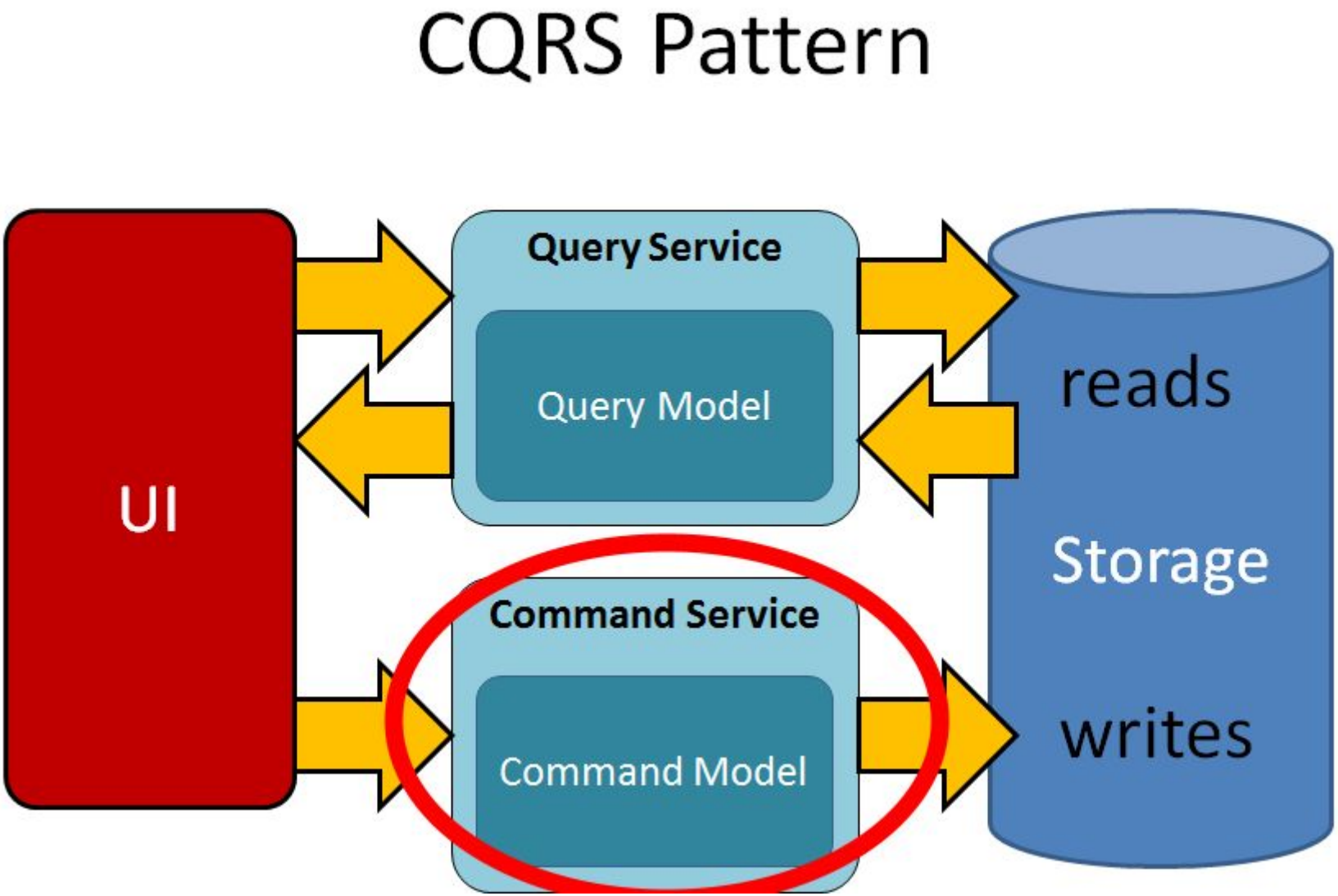


CQRS

Command-query responsibility segregation



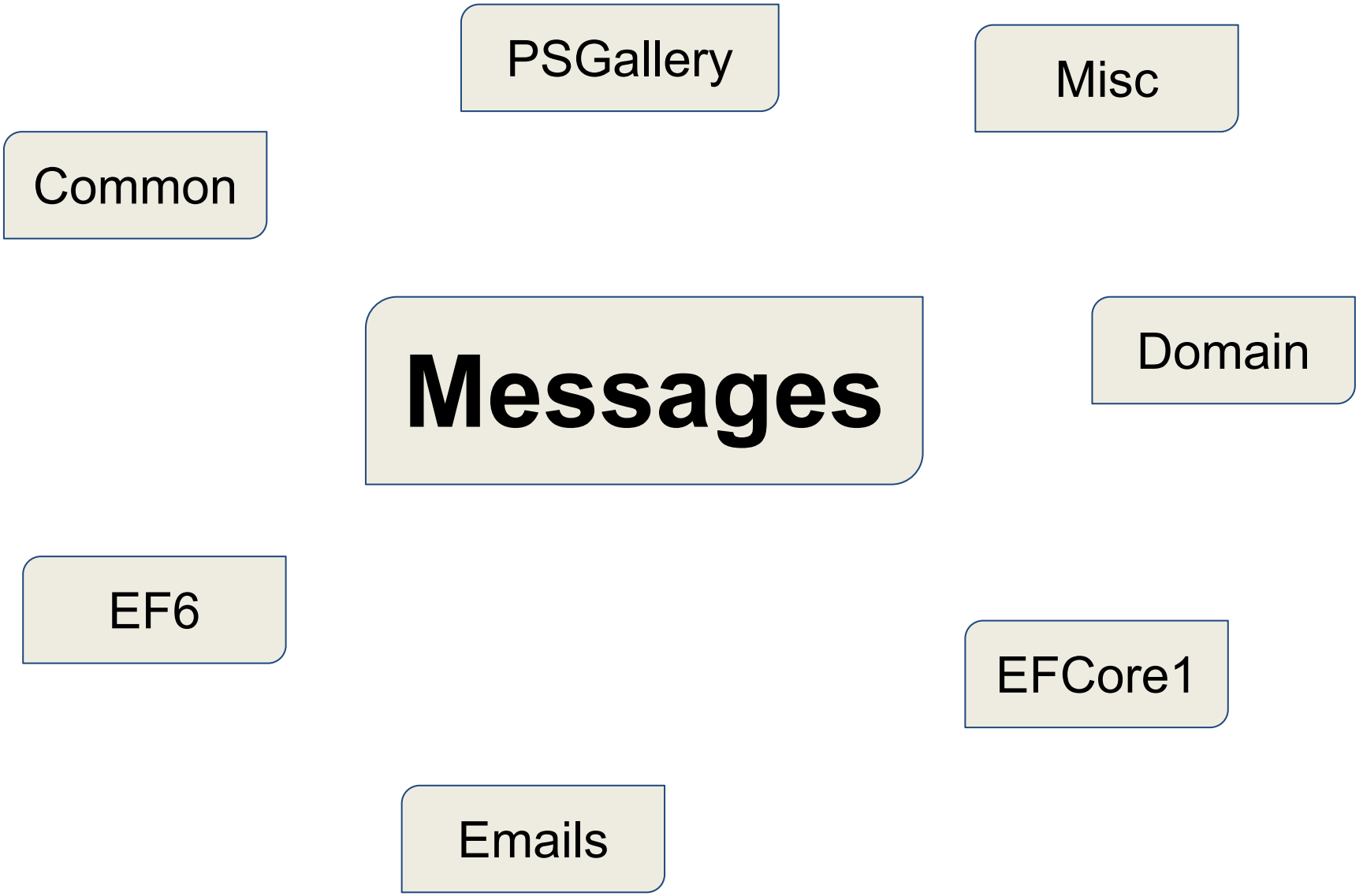
CQRS


CQRS Pattern
UI
Query Service
Query Model
Command Service
Command Model
reads
Storage
writes

Проект Saritasa Tools

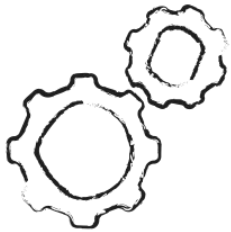
- Связывание команды и обработчика
- Сохранение списка команд
- Сохранение дополнительной информации
- ...

Saritasa Tools Messages

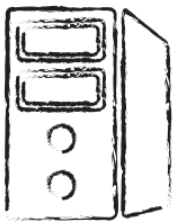


Сообщения

Пользователи



Внутренние задачи

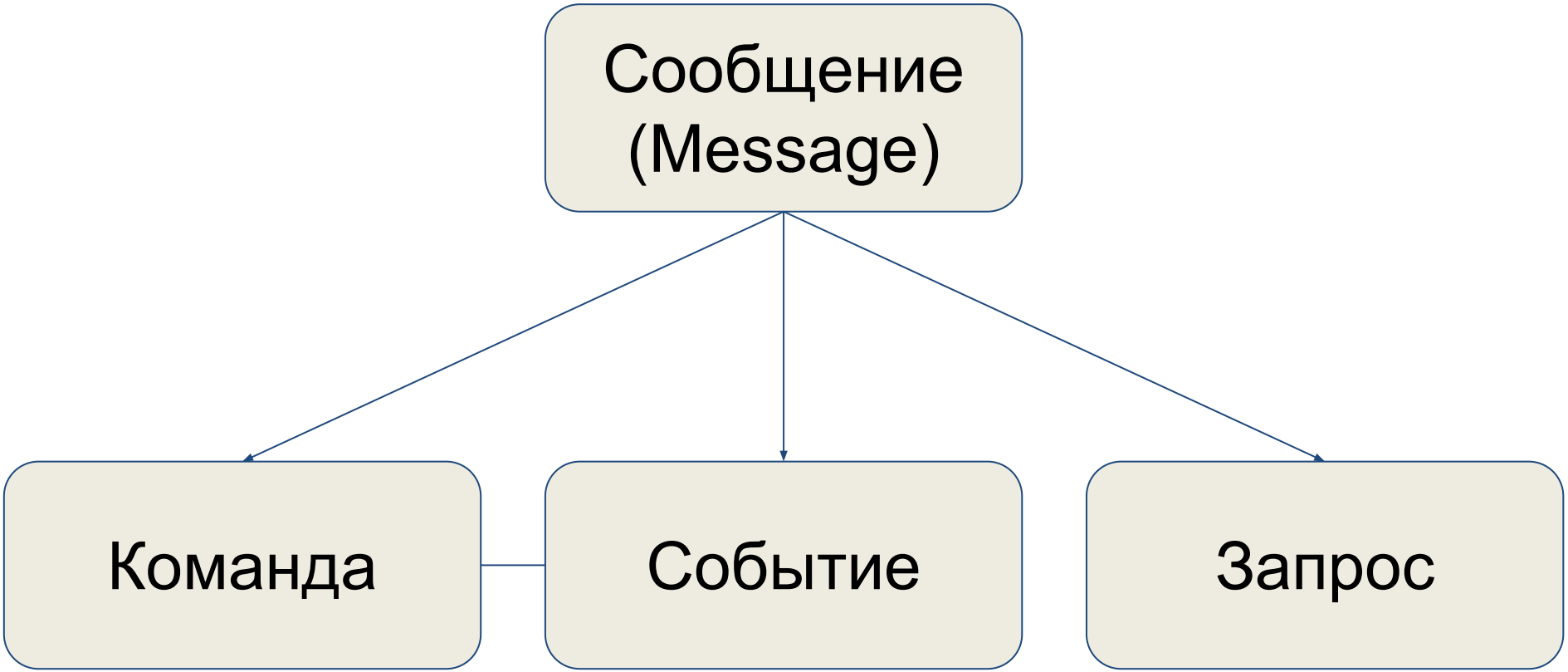


Веб-сервисы

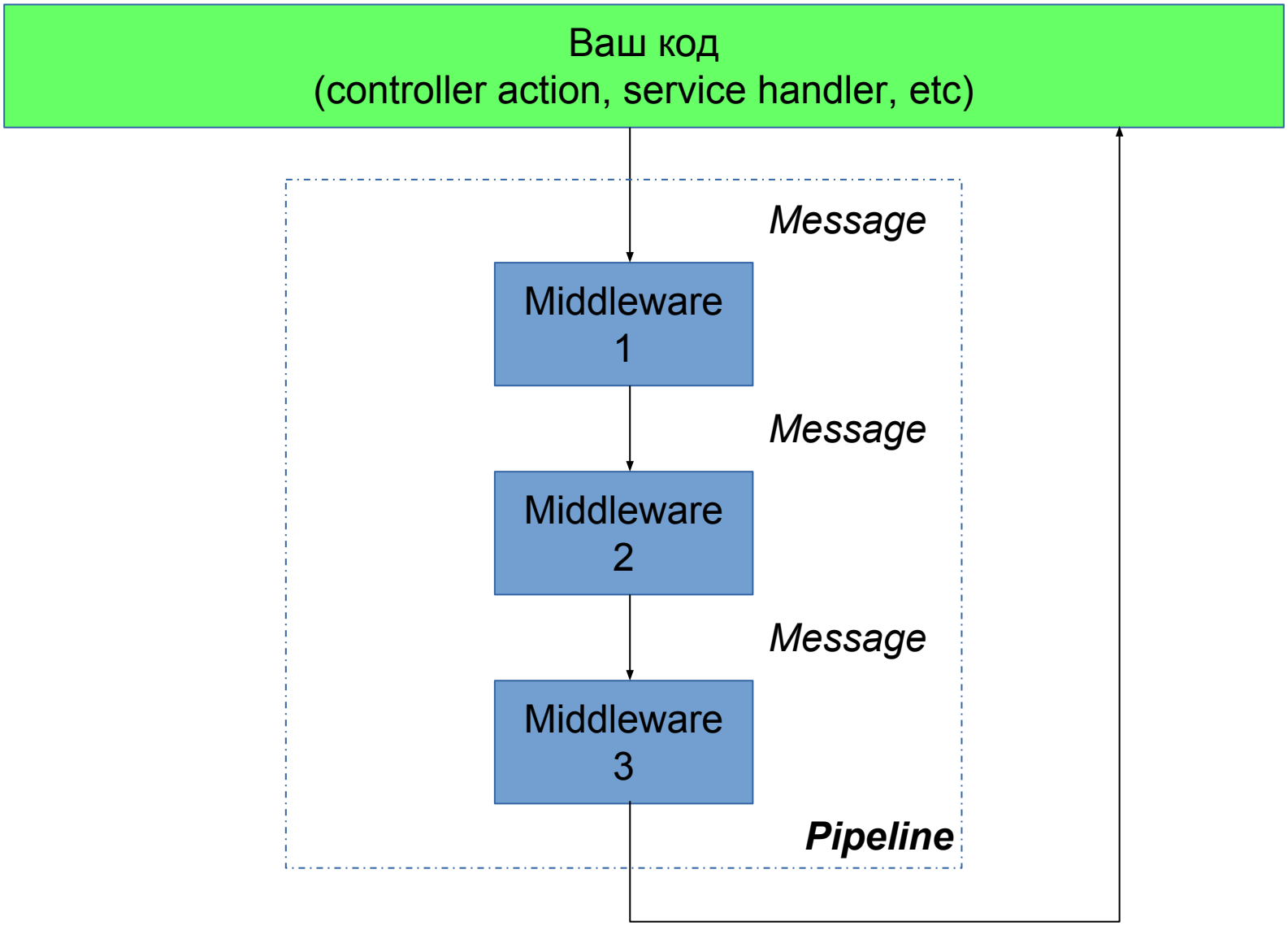
Message

Наше приложение

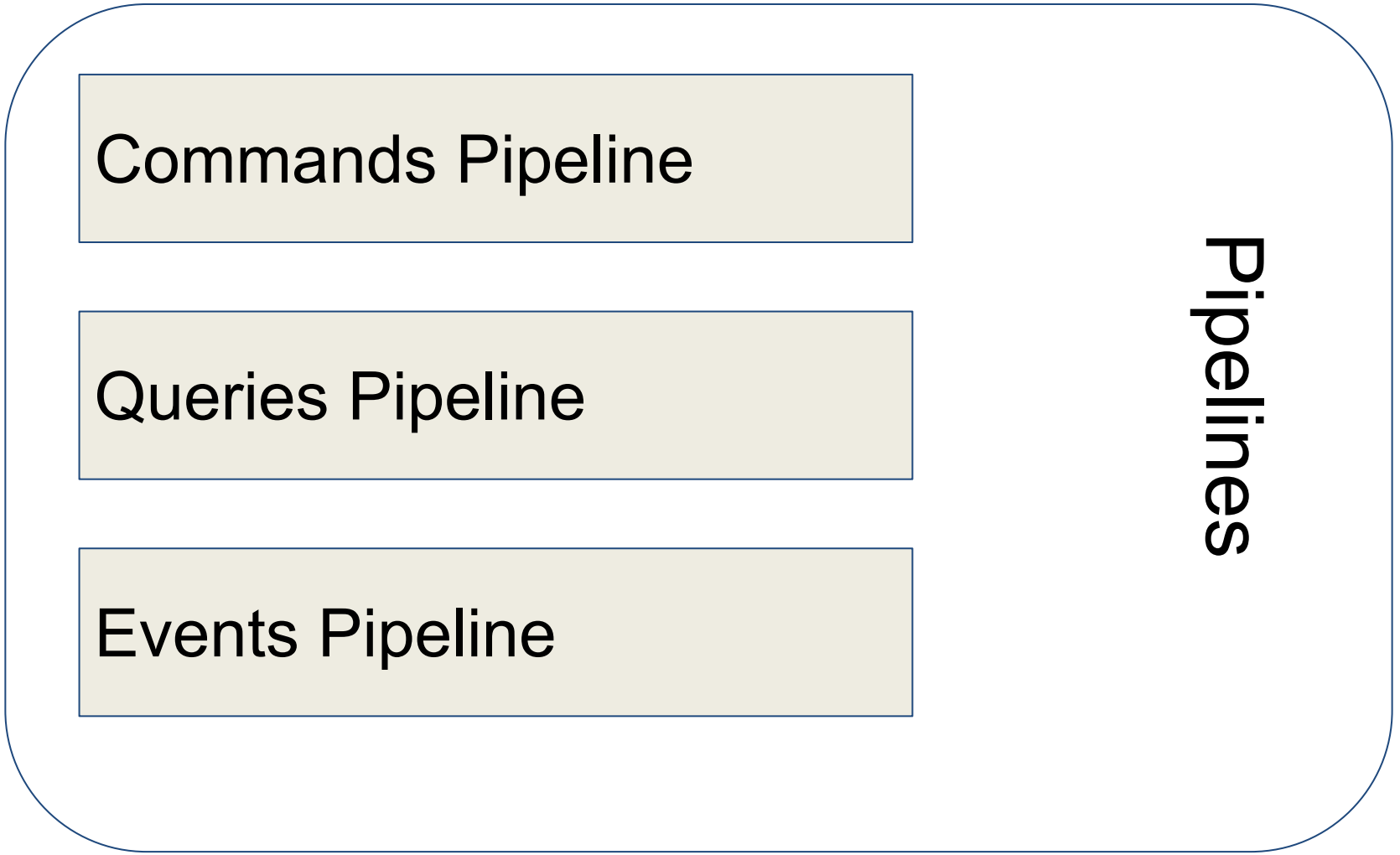
Типы сообщений



Сообщения


Ваш код
(controller action, service handler, etc)
Message
Middleware
1
Message
Middleware
2
Message
Middleware
3
Pipeline

Pipelines


Commands Pipeline
Queries Pipeline
Events Pipeline
Pipelines

Pipelines

- Commands Pipeline - Отвечает за изменение состояния приложения (например, запись в хранилище, файлы).

Pipelines

- Commands Pipeline - Отвечает за изменение состояния приложения (например, запись в хранилище, файлы).
- Query Pipeline - Используется для запросов состояния приложения.

Pipelines

- Commands Pipeline - Отвечает за изменение состояния приложения (например, запись в хранилище, файлы).
- Query Pipeline - Используется для запросов состояния приложения.
- Events Pipeline - События, происходящие в командах. Могут иметь несколько обработчиков.

Pipeline - создание

```
var container = new DefaultMessagePipelineContainer();
container.AddCommandPipeline();
```

Middleware (слой)

```
1 public class CommandHandlerLocatorMiddleware :
2     IMessagePipelineMiddleware
3 {
4     public string Id { get; } = "Locator";
5
6     public void Handle(IMessageContext messageContext)
7     {
8         // Code to find handler for command.
9     }
10 }
```

Middleware - добавление

```
1 var container = new DefaultMessagePipelineContainer();
2 container.AddCommandPipeline()
3     .AddMiddleware(new CommandHandlerLocatorMiddleware());
```

Middleware - добавление

```
1 var container = new DefaultMessagePipelineContainer();
2 container.AddCommandPipeline()
3     .AddMiddleware(new Middleware1())
4     .AddMiddleware(new Middleware2())
5     .AddMiddleware(new Middleware3());
```

Message Context

- идентификатор (GUID);
- сообщение (данные);
- статус;
- дата создания;
- исключение (если произошло);
- дополнительные данные (словарь);
- Сервис-провайдер (IServiceProvider);

Статусы сообщений

- NotInitialized;
- Processing;
- Completed;
- Failed;
- Rejected;

Repository Middleware

- Сообщения могут быть сохранены для:
  - ведения журнала;
  - повторного выполнения;
  - анализа;
- Можно установить фильтр.
- Поддерживаемые хранилища:
  - ADO.NET
  - файлы
  - Elasticsearch
  - Loggly
  - веб-сервис
  - JSON
  - ...

Формат сообщения

```
"Id": "ff369938-1e78-4dae-80dc-c22ccceef724",
```

```
1 {
2     "Id": "ff369938-1e78-4dae-80dc-c22ccceef724",
3     "Type": 1,
4     "ContentType": "SandBox.Commands.UpdateProductCommand",
5     "Content": {
6         "ProductId": 10,
7         "BestBefore": null
8     },
9     "Data": {
10        "LoggedUserId": 10
11    },
12    "ErrorType": "System.Exception",
13    "ErrorMessage": "Test",
14    "Error": {
15        "ClassName": "System.Exception",
16        "Message": "Test",
17    },
18    "CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
19    "ExecutionDuration": 14,
20    "Status": 2
21 }
```

Формат сообщения

```
"Type": 1,
```

```
1 {
2     "Id": "ff369938-1e78-4dae-80dc-c22ccceef724",
3     "Type": 1,
4     "ContentType": "SandBox.Commands.UpdateProductCommand",
5     "Content": {
6         "ProductId": 10,
7         "BestBefore": null
8     },
9     "Data": {
10        "LoggedUserId": 10
11    },
12    "ErrorType": "System.Exception",
13    "ErrorMessage": "Test",
14    "Error": {
15        "ClassName": "System.Exception",
16        "Message": "Test",
17    },
18    "CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
19    "ExecutionDuration": 14,
20    "Status": 2
21 }
```

Формат сообщения

```
"ContentType": "SandBox.Commands.UpdateProductCommand",
"Content": {
    "ProductId": 10,
    "BestBefore": null
},
```

```
 5      "Content": {
 6          "ProductId": 10,
 7          "BestBefore": null
 8      },
 9      "Data": {
10          "LoggedUserId": 10
11      },
12      "ErrorType": "System.Exception",
13      "ErrorMessage": "Test",
14      "Error": {
15          "ClassName": "System.Exception",
16          "Message": "Test",
17      },
18      "CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
19      "ExecutionDuration": 14,
20      "Status": 2
21  }
```

Формат сообщения

```
"Data": {
    "LoggedUserId": 10
},
```

```
    -4dae-80dc-c22ccceef724",
    "Type": 1,
    "ContentType": "SandBox.Commands.UpdateProductCommand",
    "Content": {
        "ProductId": 10,
        "BestBefore": null
    },
    "Data": {
        "LoggedUserId": 10
    },
    "ErrorType": "System.Exception",
    "ErrorMessage": "Test",
    "Error": {
        "ClassName": "System.Exception",
        "Message": "Test",
    },
    "CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
    "ExecutionDuration": 14,
    "Status": 2
}
```

Формат сообщения

```
"ErrorType": "System.Exception",
"ErrorMessage": "Test",
"Error": {
    "ClassName": "System.Exception",
    "Message": "Test",
},
```

```
 7          "BestBefore": null
 8      },
 9      "Data": {
10          "LoggedUserId": 10
11      },
12      "ErrorType": "System.Exception",
13      "ErrorMessage": "Test",
14      "Error": {
15          "ClassName": "System.Exception",
16          "Message": "Test",
17      },
18      "CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
19      "ExecutionDuration": 14,
20      "Status": 2
21  }
```

Формат сообщения

```
"CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
```

```
1 {
2     "Id": "ff369938-1e78-4dae-80dc-c22ccceef724",
3     "Type": 1,
4     "ContentType": "SandBox.Commands.UpdateProductCommand",
5     "Content": {
6         "ProductId": 10,
7         "BestBefore": null
8     },
9     "Data": {
10        "LoggedUserId": 10
11    },
12    "ErrorType": "System.Exception",
13    "ErrorMessage": "Test",
14    "Error": {
15        "ClassName": "System.Exception",
16        "Message": "Test",
17    },
18    "CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
19    "ExecutionDuration": 14,
20    "Status": 2
21 }
```

Формат сообщения

```
"ExecutionDuration": 14,
```

```
1 {
2     "Id": "ff369938-1e78-4dae-80dc-c22ccceef724",
3     "Type": 1,
4     "ContentType": "SandBox.Commands.UpdateProductCommand",
5     "Content": {
6         "ProductId": 10,
7         "BestBefore": null
8     },
9     "Data": {
10        "LoggedUserId": 10
11    },
12    "ErrorType": "System.Exception",
13    "ErrorMessage": "Test",
14    "Error": {
15        "ClassName": "System.Exception",
16        "Message": "Test",
17    },
18    "CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
19    "ExecutionDuration": 14,
20    "Status": 2
21 }
```

Формат сообщения

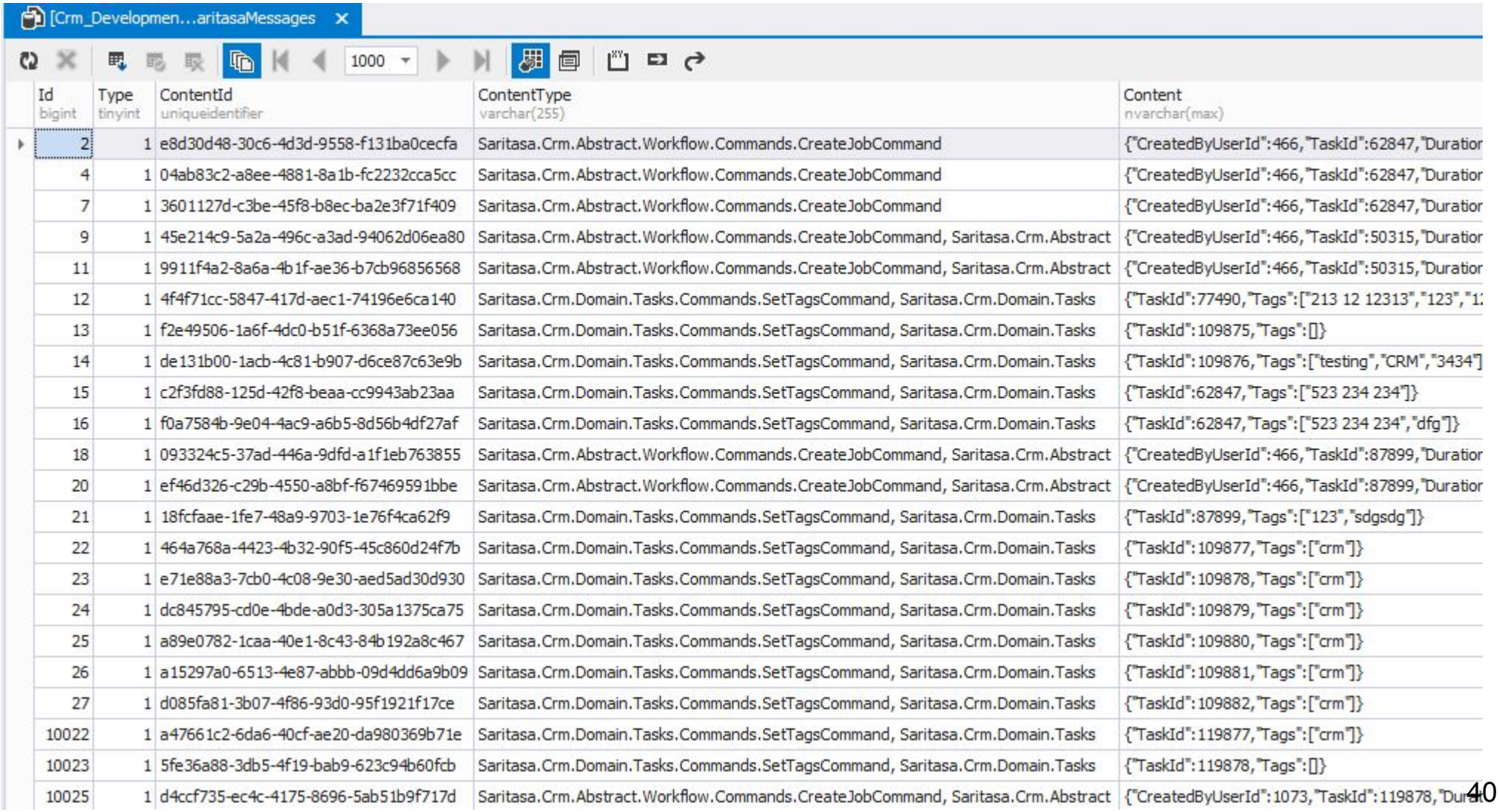
```
"Status": 2
```

```
{
    "Id": "ff369938-1e78-4dae-80dc-c22ccceef724",
    "Type": 1,
    "ContentType": "SandBox.Commands.UpdateProductCommand",
    "Content": {
        "ProductId": 10,
        "BestBefore": null
    },
    "Data": {
        "LoggedUserId": 10
    },
    "ErrorType": "System.Exception",
    "ErrorMessage": "Test",
    "Error": {
        "ClassName": "System.Exception",
        "Message": "Test",
    },
    "CreatedAt": "2016-10-11T21:12:31.4461153+07:00",
    "ExecutionDuration": 14,
    "Status": 2
}
```

Repository Middleware - Инициализация

```
1 var connectionString = ConfigurationManager
2     .ConnectionStrings["DbContext"];
3
4 var repositoryMiddleware = new RepositoryMiddleware(
5     new Repositories.AdoNetMessageRepository(
6         DbProviderFactories.GetFactory(connectionString.ProviderName),
7         connectionString)
8 );
9
10 pipelinesContainer.AddQueryPipeline()
11     .AddMiddleware(repositoryMiddleware);
```

ADO.NET Repository



[Crm_Developmen...aritasaMessages

1000

Id bigint	Type tinyint	ContentId uniqueidentifier	ContentType varchar(255)	Content nvarchar(max)
2	1	e8d30d48-30c6-4d3d-9558-f131ba0cecfa	Saritasa.Crm.Abstract.Workflow.Commands.CreateJobCommand	{"CreatedByUserId":466,"TaskId":62847,"Duration
4	1	04ab83c2-a8ee-4881-8a1b-fc2232cca5cc	Saritasa.Crm.Abstract.Workflow.Commands.CreateJobCommand	{"CreatedByUserId":466,"TaskId":62847,"Duration
7	1	3601127d-c3be-45f8-b8ec-ba2e3f71f409	Saritasa.Crm.Abstract.Workflow.Commands.CreateJobCommand	{"CreatedByUserId":466,"TaskId":62847,"Duration
9	1	45e214c9-5a2a-496c-a3ad-94062d06ea80	Saritasa.Crm.Abstract.Workflow.Commands.CreateJobCommand, Saritasa.Crm.Abstract	{"CreatedByUserId":466,"TaskId":50315,"Duration
11	1	9911f4a2-8a6a-4b1f-ae36-b7cb96856568	Saritasa.Crm.Abstract.Workflow.Commands.CreateJobCommand, Saritasa.Crm.Abstract	{"CreatedByUserId":466,"TaskId":50315,"Duration
12	1	4f4f71cc-5847-417d-aec1-74196e6ca140	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":77490,"Tags":["213 12 12313","123","1
13	1	f2e49506-1a6f-4dc0-b51f-6368a73ee056	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":109875,"Tags":[]}
14	1	de131b00-1acb-4c81-b907-d6ce87c63e9b	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":109876,"Tags":["testing","CRM","3434"]
15	1	c2f3fd88-125d-42f8-beaa-cc9943ab23aa	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":62847,"Tags":["523 234 234"]}
16	1	f0a7584b-9e04-4ac9-a6b5-8d56b4df27af	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":62847,"Tags":["523 234 234","dfg"]}
18	1	093324c5-37ad-446a-9dfd-a1f1eb763855	Saritasa.Crm.Abstract.Workflow.Commands.CreateJobCommand, Saritasa.Crm.Abstract	{"CreatedByUserId":466,"TaskId":87899,"Duration
20	1	ef46d326-c29b-4550-a8bf-f67469591bbe	Saritasa.Crm.Abstract.Workflow.Commands.CreateJobCommand, Saritasa.Crm.Abstract	{"CreatedByUserId":466,"TaskId":87899,"Duration
21	1	18fcfaae-1fe7-48a9-9703-1e76f4ca62f9	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":87899,"Tags":["123","sdgsdg"]}
22	1	464a768a-4423-4b32-90f5-45c860d24f7b	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":109877,"Tags":["crm"]}
23	1	e71e88a3-7cb0-4c08-9e30-aed5ad30d930	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":109878,"Tags":["crm"]}
24	1	dc845795-cd0e-4bde-a0d3-305a1375ca75	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":109879,"Tags":["crm"]}
25	1	a89e0782-1caa-40e1-8c43-84b192a8c467	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":109880,"Tags":["crm"]}
26	1	a15297a0-6513-4e87-abbb-09d4dd6a9b09	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":109881,"Tags":["crm"]}
27	1	d085fa81-3b07-4f86-93d0-95f1921f17ce	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":109882,"Tags":["crm"]}
10022	1	a47661c2-6da6-40cf-ae20-da980369b71e	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":119877,"Tags":["crm"]}
10023	1	5fe36a88-3db5-4f19-bab9-623c94b60fcb	Saritasa.Crm.Domain.Tasks.Commands.SetTagsCommand, Saritasa.Crm.Domain.Tasks	{"TaskId":119878,"Tags":[]}
10025	1	d4ccf735-ec4c-4175-8696-5ab51b9f717d	Saritasa.Crm.Abstract.Workflow.Commands.CreateJobCommand, Saritasa.Crm.Abstract	{"CreatedByUserId":1073,"TaskId":119878,"Dura

Middleware Repositories - фильтрация

```
1 var filter = RepositoryMessagesFilter
2             .Create()
3             .WithExecutionDurationAbove(300)
4             .WithExcludeContentType(new Regex(@"\.PingCommand$"));
5
6 var repositoryMiddleware = new RepositoryMiddleware(
7     adoNetRepository,
8     filter
9 );
```

Command (команда)

- Изменяет состояние приложения.
- Команда и обработчик — разные сущности.
- Может возвращать результат.

Пример команды

```
1 public class CreateTaskCommand
2 {
3     public string UserId { get; set; }
4
5     public int ProjectId { get; set; }
6
7     [MaxLength(255)]
8     public string Text { get; set; }
9
10    public bool IsDone { get; set; }
11
12    [DataType(DataType.Date)]
13    public DateTime? DueDate { get; set; }
14
15    [CommandOut]
16    public int TaskId { get; set; }
17 }
```

Handler (обработчик)

```
1 [CommandHandlers]
2 public class TaskHandlers
3 {
4     public void HandleCreateTask(CreateTaskCommand command,
5         IAppUnitOfWorkFactory uowFactory)
6     {
7         using (var uow = uowFactory.Create())
8         {
9             // Do something useful!
10        }
11    }
12 }
```

Регистрация обработчиков

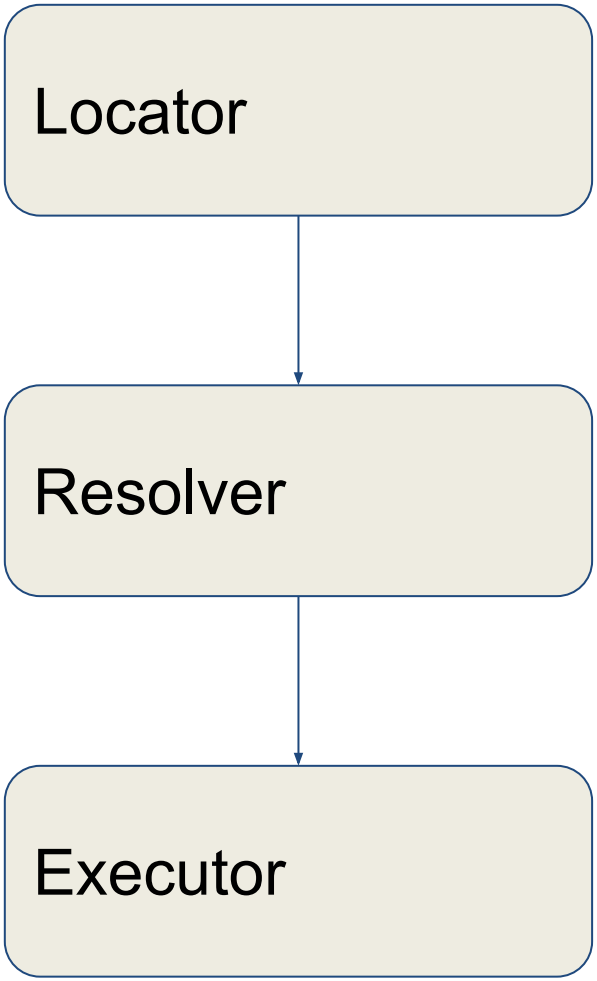
```
1 IMessagePipelineContainer pipelinesContainer =
2     new DefaultMessagePipelineContainer();
3 pipelinesContainer.AddCommandPipeline()
4     .AddMiddleware(new CommandHandlerLocatorMiddleware(
5             Assembly.GetAssembly(typeof(Entities.User))))
6     .AddMiddleware(new CommandHandlerResolverMiddleware())
7     .AddMiddleware(new CommandHandlerExecutorMiddleware
8     {
9         UseParametersResolve = true
10    })
11    .AddMiddleware(adoNetRepositoryMiddleware);
```

Выполнение

```
1 public class TaskController : Controller
2 {
3     private IMessagePipelineService pipelineService;
4
5     public TaskController(IMessagePipelineService pipelineService)
6         : base(pipelineService) =>
7             this.pipelineService = pipelineService;
8
9     [HttpPost]
10    public ActionResult Post(CreateTaskCommand command)
11    {
12        pipelineService.HandleCommand(command);
13        return Json(command.TaskId);
14    }
15 }
```

Выполнение

Поиск pipeline, формирование контекста сообщения и передача первому слою.



Поиск обработчика

Создание класса обработчика

Выполнение

Внедрение зависимостей

```
1  public void HandleCreateTask(CreateTaskCommand command,
2      IAppUnitOfWorkFactory uowFactory)
3  {
4      using (var uow = uowFactory.Create())
5      {
6          // ...
7
8          uow.SaveChanges();
9      }
10 }
```

Внедрение зависимостей

- IMessagePipelineContainer - Singleton
- IMessagePipelineService - Transient (Scoped)
  - Pipeline Container
  - IServiceProvider

- Container
- Service Provider

Внедрение зависимостей

```
1 public interface IServiceProvider
2 {
3     object GetService(Type serviceType);
4 }
5
6 MessageContext.ServiceProvider
```

Внедрение зависимостей

```
1 var pipelineContainer = new DefaultMessagePipelineContainer();
2 services.AddSingleton<IMessagePipelineContainer>(pipelineContainer);
3 services.AddScoped<IMessagePipelineService, DefaultMessagePipelineService>();
```

Query Pipeline

1. Не изменяет состояние.
2. Рекомендуется использовать в том случае, если необходимо журналировать аргументы метода и результат.

Query Pipeline

```
 1 [QueryHandlers]
 2 public class TasksQueries
 3 {
 4     public TaskDto GetByIdDto(int taskId) =>
 5         new TaskDto(uow.TaskRepository.Get(taskId));
 6 }
 7
 8 // Controller.
 9 var data = pipelineService.Query<TasksQueries>()
10     .With(q => q.GetByIdDto(userId));
```

Query Pipeline

```
1 [QueryHandlers]
2 public class TasksQueries
3 {
4     public TaskDto GetByIdDto(int taskId) =>
5         new TaskDto(uow.TaskRepository.Get(taskId));
6 }
7
8 // Controller.
9 var data = pipelineService.Query<TasksQueries>()
10     .With(q => q.GetByUserDto(userId));
```

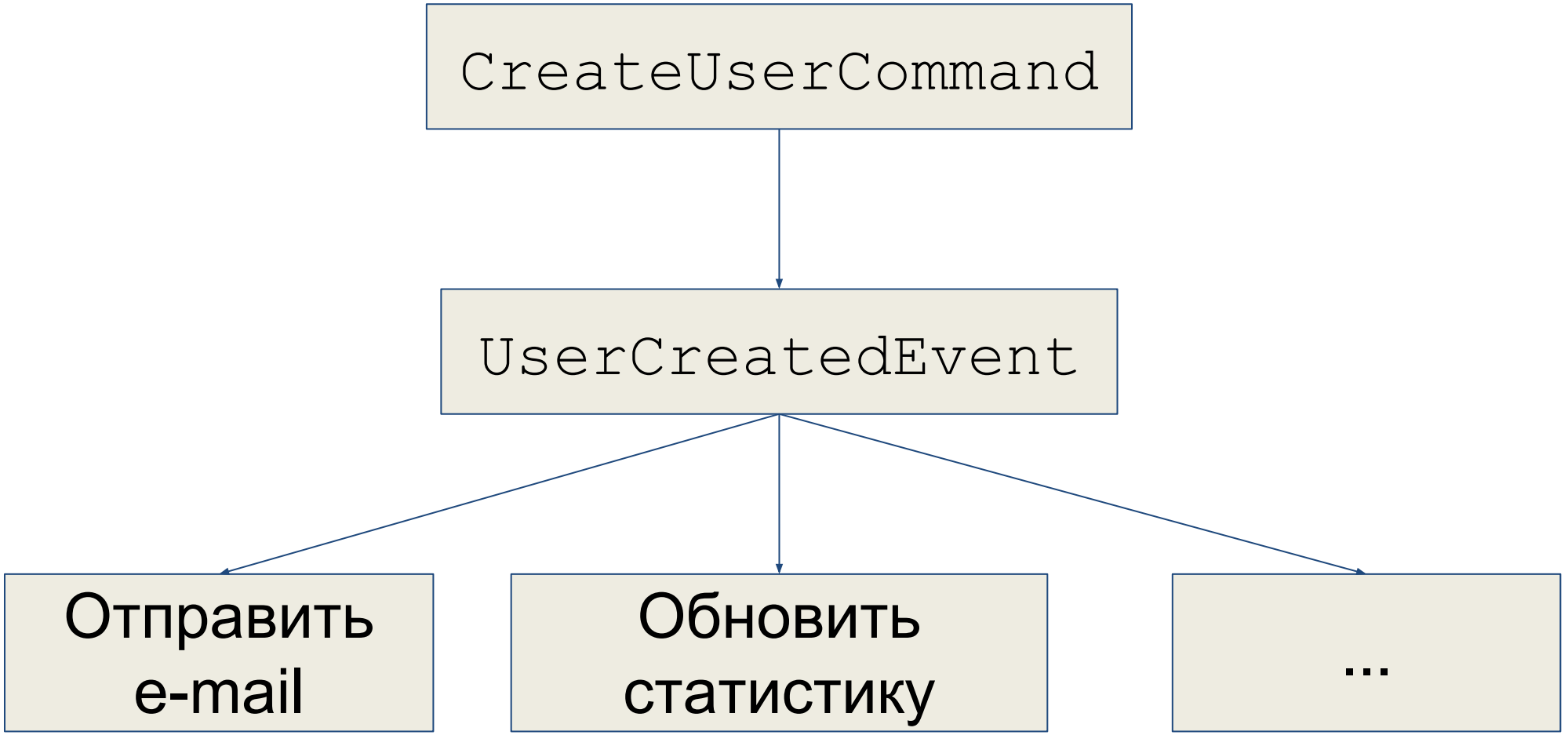
Query Pipeline - реализация

```
1 public interface IQueryPipeline : IMessagePipeline
2 {
3     IQueryCaller<TQuery> Query<TQuery>(
4         IMessagePipelineService pipelineService)
5             where TQuery : class;
6 }
```

Query Pipeline - реализация

```
1 public interface IQueryCaller<TQuery>
2     where TQuery : class
3 {
4     TResult With<TResult>(
5         Expression<Func<TQuery, TResult>> expression);
6 }
```

Events Pipeline



Events Pipeline - Event Class

```
1 // Event.
2 public class UserCreatedEvent
3 {
4     public User User { get; set; }
5 }
```

Events Pipeline - Event Handler

```
1 // Handler.
2 [EventHandlers]
3 public class UserEventsHandlers
4 {
5     public void HandleSendEmailOnUserCreate(
6         UserCreatedEvent userCreatedEvent)
7     {
8         // Code
9     }
10 }
```

Events Pipeline - Raise Event

```
1 // Raise.
2 pipelineService.RaiseEvent(new UserCreatedEvent
3 {
4     User = user
5 });
```

Собственный middleware

```
1 public class UserMemoryMiddleware : IMessagePipelineMiddleware
2 {
3     public string Id { get; } = "UserMemoryMiddleware";
4
5     public void Handle(IMessageContext messageContext)
6     {
7         var data = new Dictionary<string, string>
8         {
9             ["LoggedUserId"] = HttpHelper.LoggedUserId,
10            ["MemoryUsage"] =
11                Process.GetCurrentProcess().WorkingSet64.ToString()
12        };
13        messageContext.Items[".data"] = data;
14    }
15 }
```

Производительность

1. Замена рефлексии кодогенерацией.
2. Кэширование.

Производительность

```
1 public void Handle(CreateCommand command, IUow uow)
2 {}
```

Производительность

```
1 public void Handle(CreateCommand command, IUow uow)
2 {}
```



```
1 Expression<Func<IServiceProvider, object>>
```



```
1 .Block(CommandsTests+Ns07_TestCommandHandlers3 $handler) {
2     $handler = .New CommandsTests+Ns07_TestCommandHandlers3(
3             (CommandsTests+Ns01_IInterfaceA)
4         .Call $sp.GetService(.Constant<System.RuntimeType>
5             (CommandsTests+Ns01_IInterfaceA))
6     );
7     $handler
8 }
```

Производительность

Кеширование делегатов дало увеличение производительности выполнения команд примерно в два раза.

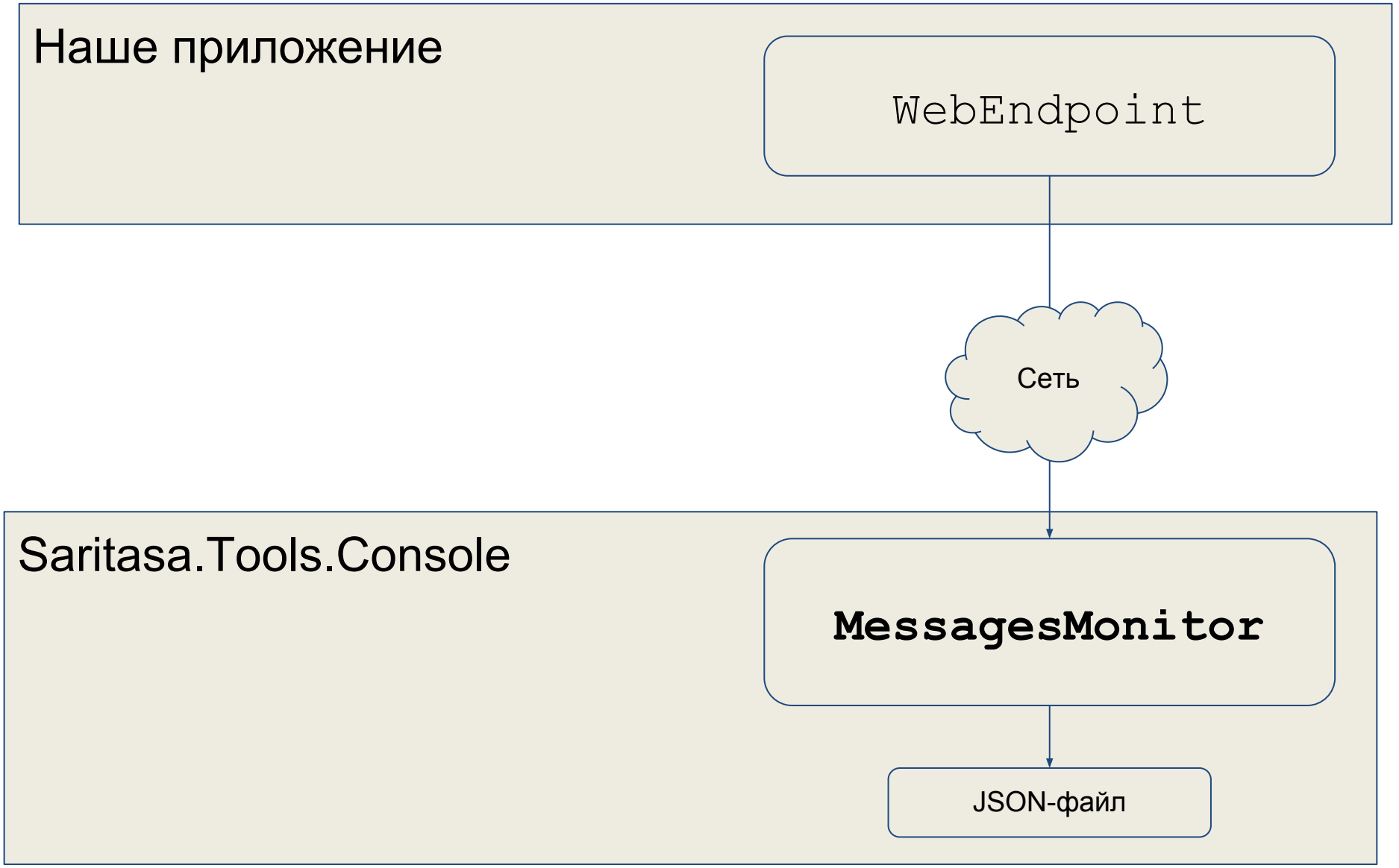
Было

```
BenchmarkDotNet=v0.10.8, OS=Windows 10 Redstone 2 (10.0.15063)
Processor=Intel Core i5-6400 CPU 2.70GHz (Skylake), ProcessorCount=4
Frequency=2648433 Hz, Resolution=377.5818 ns, Timer=TSC
  [Host]     : Clr 4.0.30319.42000, 32bit LegacyJIT-v4.7.2115.0
  DefaultJob : Clr 4.0.30319.42000, 32bit LegacyJIT-v4.7.2115.0


                  Method |       Mean |     Error |    StdDev | Scaled | ScaledSD |
---------------------- |-----------:|----------:|----------:|-------:|---------:|
       RunCommandDirect |   2.201 ms | 0.0308 ms | 0.0288 ms |   1.00 |     0.00 |
 RunCommandWithPipeline | 172.551 ms | 0.9631 ms | 0.9008 ms |  78.40 |     1.06 |
```

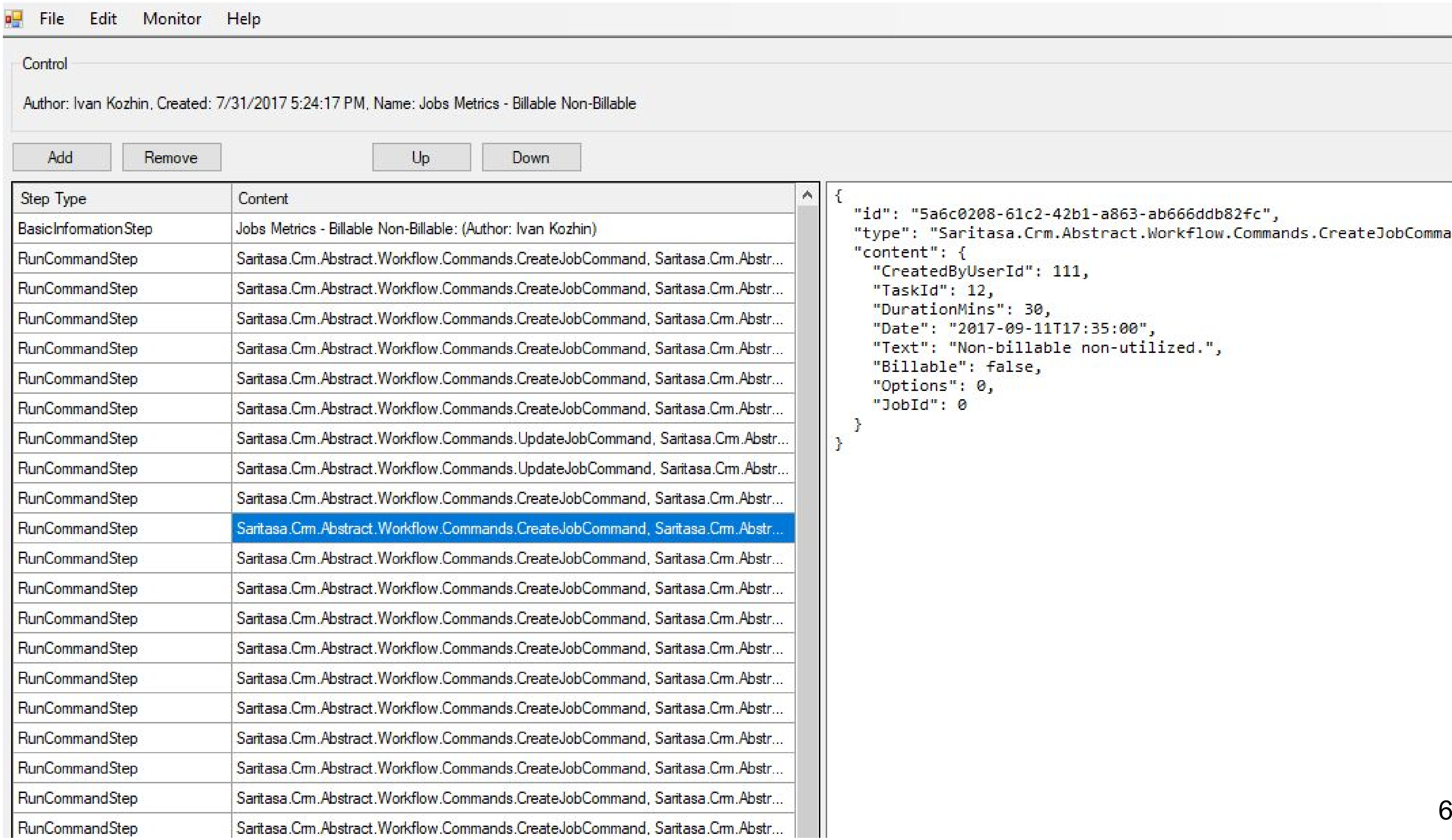
Стало

```
BenchmarkDotNet=v0.10.9, OS=Windows 10 Redstone 2 (10.0.15063)
Processor=Intel Core i5-6400 CPU 2.70GHz (Skylake), ProcessorCount=4
Frequency=2648436 Hz, Resolution=377.5813 ns, Timer=TSC
  [Host]     : .NET Framework 4.7 (CLR 4.0.30319.42000), 32bit LegacyJIT-v4.7.2115.0
  DefaultJob : .NET Framework 4.7 (CLR 4.0.30319.42000), 32bit LegacyJIT-v4.7.2115.0


                  Method |      Mean |     Error |    StdDev | Scaled | ScaledSD |
---------------------- |----------:|----------:|----------:|-------:|---------:|
       RunCommandDirect |  2.338 ms | 0.0453 ms | 0.0424 ms |   1.00 |     0.00 |
 RunCommandWithPipeline | 98.151 ms | 0.6594 ms | 0.6168 ms |  41.99 |     0.77 |
```

Использование в тестировании



Использование в тестировании

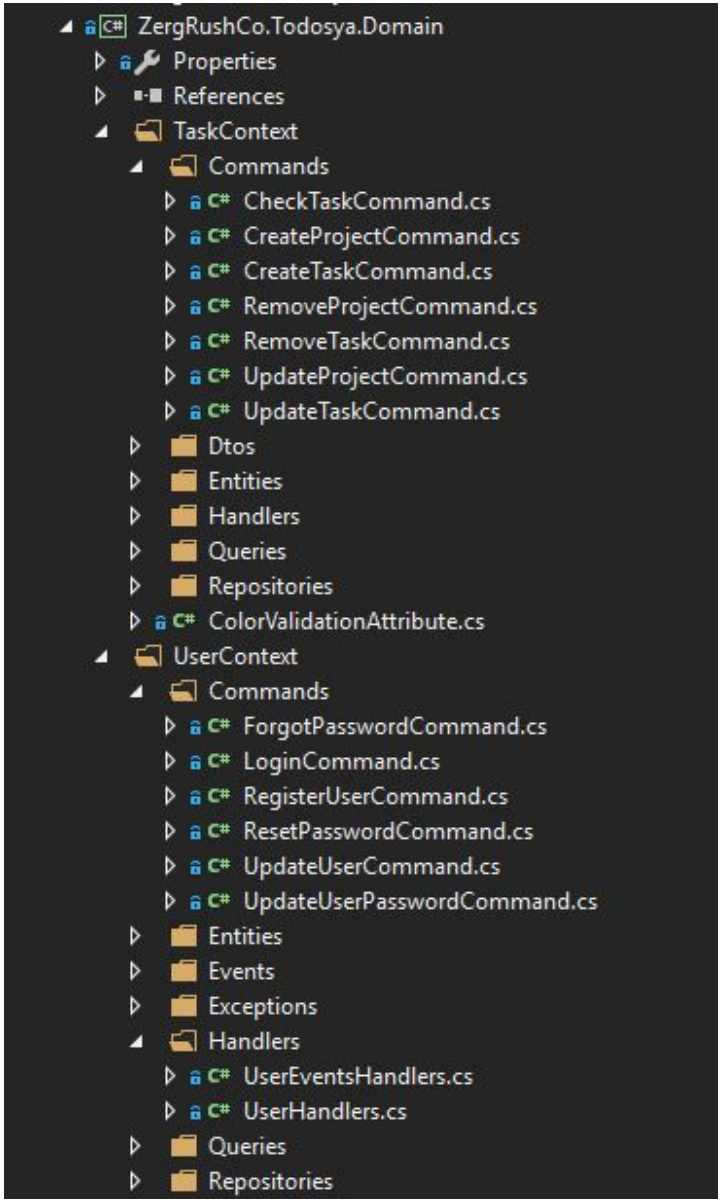


Лог сообщений

~Event Sourcing

Структура проекта

- Domain
  - Commands
  - Entities
  - Events
  - Handlers
  - Queries
  - ...



А как же “легаси”?

“Легаси”

1. Не всё так просто, зависит от проекта.

“Легаси”

1. Не всё так просто, зависит от проекта.
2. Можно подключить библиотеку и использовать там, где журналирование наиболее необходимо.

“Легаси”

1. Не всё так просто, зависит от проекта.
2. Можно подключить библиотеки и использовать там, где журналирование наиболее необходимо.
3. Переделывать при рефакторинге.

Что получилось?

- Унифицировать архитектуру проектов.

Что получилось?

- Унифицировать архитектуру проектов.
- Упростить отладку при возникновении сбоев.

Что получилось?

- Унифицировать архитектуру проектов.
- Упростить отладку при возникновении сбоев.
- Ввести интеграционное тестирование на основе команд.

Что получилось?

- Унифицировать архитектуру проектов.
- Упростить отладку при возникновении сбоев.
- Ввести интеграционное тестирование на основе команд.
- Осуществлять простой мониторинг приложения (больше данных о том, что происходит внутри).

Над чем надо работать

- Всё ещё бета.

Над чем надо работать

- Всё ещё бета.
- Нет нормальных средств для сбора статистики.

Над чем надо работать

- Всё ещё бета.
- Нет нормальных средств для сбора статистики.
- Интерфейс пользователя для просмотра сообщений в разработке.

Над чем надо работать

- Всё ещё бета.
- Нет нормальных средств для сбора статистики.
- Интерфейс пользователя для просмотра сообщений в разработке.
- Всё ещё находятся баги.

Над чем надо работать

- Всё ещё бета.
- Нет нормальных средств для сбора статистики.
- Интерфейс пользователя для просмотра сообщений в разработке.
- Всё ещё находятся баги.
- Документация :)

И еще один вывод

Фреймворк - это сложно

Кто помогал...





Спасибо за внимание!

- https://github.com/Saritasa/SaritasaTools
- https://saritasa-tools.readthedocs.io/en/latest/