# Система DLP

Черновик

## 1. Введение

Документ описывает общую концепцию DLP-систем, приблизительные принципы их работы и способы взаимодействия с пользователями, администраторами и сотрудниками отдела информационной безопасности.

Система *DLP* (англ. *Data Leak Prevention*) - специальная информационная система, созданная для предотвращения утечек конфиденциальной информации вовне. DLP-системы строятся на принципах анализа потоков данных, пересекающих периметр защищаемой информационной системы. При детектировании в этом потоке конфиденциальной информации срабатывает активная компонента системы, и передача сообщения блокируется.

Распознавание конфиденциальной информации в *DLP*-системах может производиться различными способами: анализом формальных признаков посредством строгих алгоритмов (регулярных выражений, метрик, шаблонов, т.п.) и применением нечеткой логики (методов машинного обучения, семантического анализа текстов). Первый способ позволяет избежать ложных срабатываний, но зато требует предварительной классификации документов, внедрения меток, сбора сигнатур и т.д. Второй способ предполагает ложные срабатывания, но потенциально способен обнаруживать трудноформализуемые угрозы.

### 1.1. Цель

*Указать цель создания документа.*

### 1.2. Область действия

*Краткое описание области действия документа: с какими проектами и документами он связан, все остальное, что может повлиять на документ и на что он сам влияет.*

### 1.3. Ссылки

*Полный список всех документов, на которые есть ссылки в других частях данного документа.*

### 1.4. Обзор

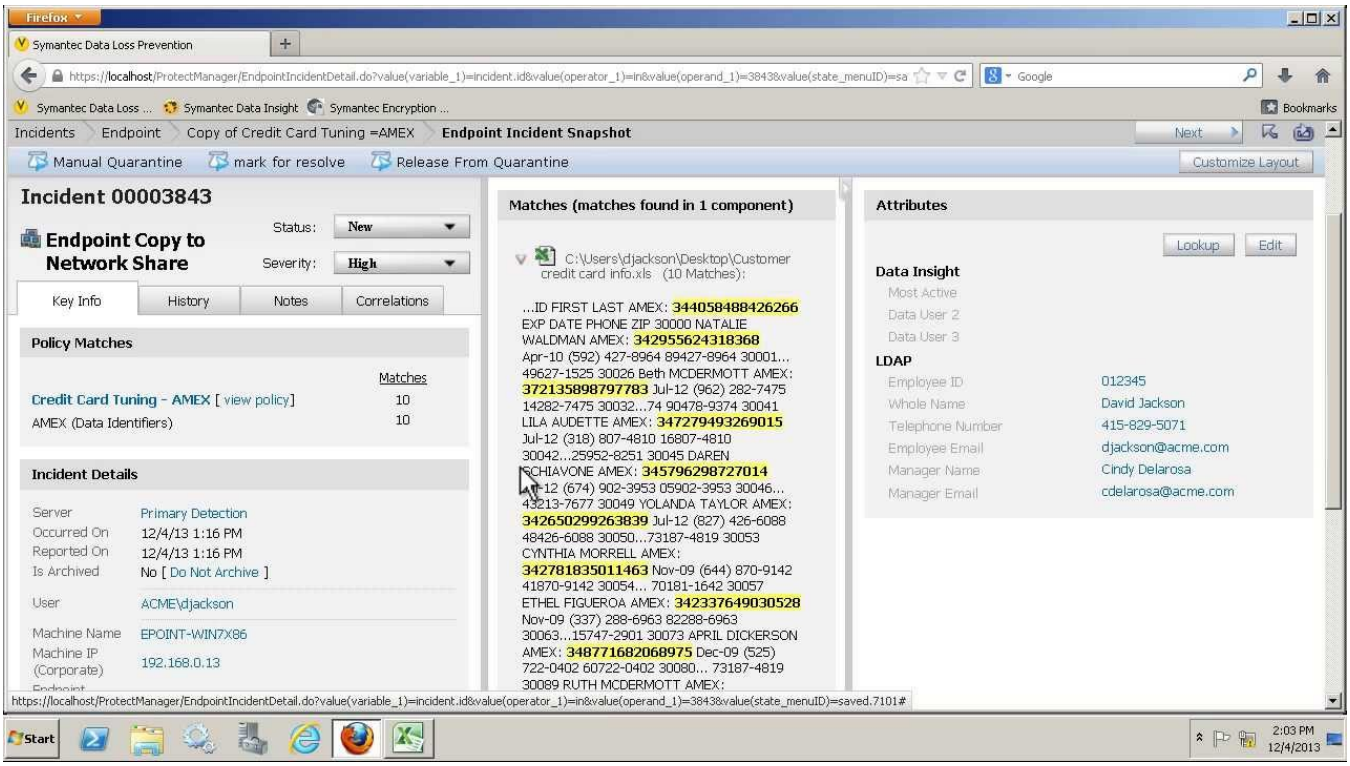
*Описание того, что содержится в остальной части документа и объяснение, как документ организован.*

## 2. Определения

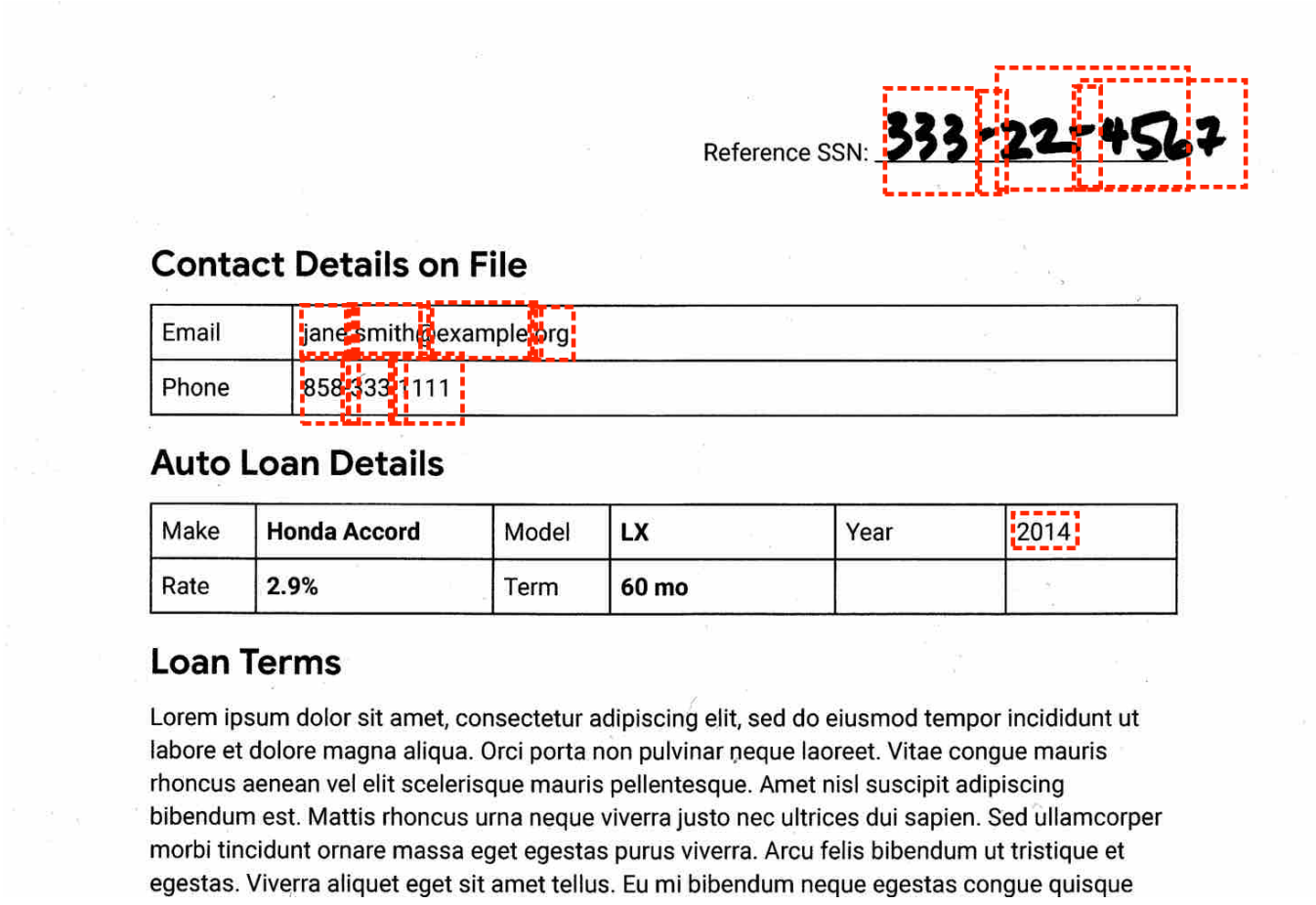
*Перечень определений и сокращений, присутствующих в тексте. Могут быть представлены в любом удобном порядке.*

## 3. Концепция DLP

Пример срабатывания DLP от Symantec на файле Excel. Видно, как в превью файла цветом выделены фрагменты текста, связанные со сработавшими правилами проверки.




А это пример превью скана бумажного документа или PDF-документа. Красными рамками выделены распознанные фрагменты с приватными данными. Очевидно, DLP-системы содержат OCR-модуль для таких целей.



Reference SSN: 333-22-4567

## Contact Details on File

| | |
|---|---|
| Email | jane.smith@example.org |
| Phone | 858.333.1111 |

## Auto Loan Details

| | | | | | |
|---|---|---|---|---|---|
| Make | **Honda Accord** | Model | **LX** | Year | 2014 |
| Rate | **2.9%** | Term | **60 mo** | | |

## Loan Terms

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Orci porta non pulvinar neque laoreet. Vitae congue mauris rhoncus aenean vel elit scelerisque mauris pellentesque. Amet nisl suscipit adipiscing bibendum est. Mattis rhoncus urna neque viverra justo nec ultrices dui sapien. Sed ullamcorper morbi tincidunt ornare massa eget egestas purus viverra. Arcu felis bibendum ut tristique et egestas. Viverra aliquet eget sit amet tellus. Eu mi bibendum neque egestas congue quisque

# 4. API

## 4.1. Контракт

Ниже представлено приблизительное представление о том, как должен выглядеть контракт адаптера DLP-системы. Сорян, это псевдокод, но вроде и так понятно.

Прикладного программиста прежде всего интересует API для анализа содержимого (простого текста, документов MS Office или PDF). Интерфейс `IDataLossPreventor` позволяет проанализировать переданное содержимое и получить в ответ коллекцию правил, которые сработали на этом содержимом.

```
        /// <summary>
        ///     .
        /// </summary>
        public interface IDataLossPreventor
        {
                /// <summary>
                ///       .
                ///   ,     ,
                ///      ,    .
                /// </summary>
                /// <remarks>
                ///      ,     DLP-   ,
                ///             .
                /// </remarks>
                /// <param name="applicationID">  ,   DLP-.</param>
                /// <param name="content">  .</param>
                /// <returns>   .</returns>
                IEnumerable<Policy> Inspect(string applicationID, Content content);
        }
```

Метод `Inspect` интерфейса `IDataLossPreventor` возвращает коллекцию экземпляров класса `Policy`:

```
        /// <summary>
        ///   .
        /// </summary>
        public sealed class Policy
        {
            /// <summary>
            ///   ,   DLP-.
            /// </summary>
            public string Name { get; set; }

            /// <summary>
            ///       DLP-.
            /// </summary>
        public string Message { get; set; }

        /// <summary>
        ///  ,      .
        /// </summary>
        public Similarity Similarity { get; set; }

        /// <summary>
        ///  ,      .
        /// </summary>
        public Similarity Tolerance { get; set; }

        /// <summary>
        ///   ,     ,    .
        /// </summary>
        public string ContentID { get; set; }

        /// <summary>
        ///    ,     .
        /// </summary>
        public IEnumerable<Location> Locations { get; set; }
    }
```

На при вызове методу `Inspect` передается экземпляр класса `Content`, представляющий содержимое анализируемого документа, а так же (желательно) тип этого содержимого.

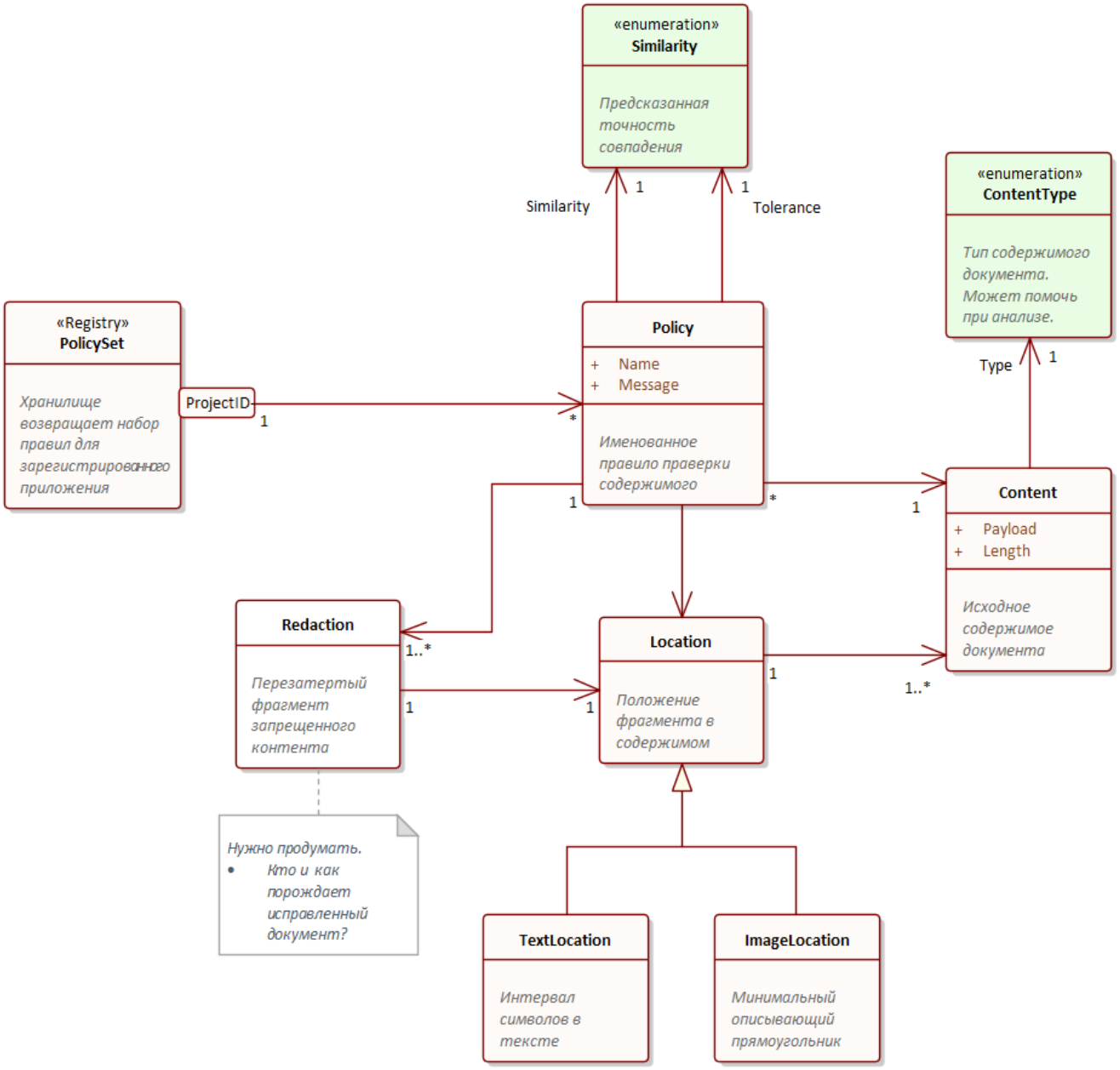
```
    /// <summary>
///   .
/// </summary>
public sealed class Content
{
    /// <summary>
    ///   .
    ///      .
    /// </summary>
    public string ContentID { get; set; }

    /// <summary>
    /// .
           ///   .
    /// </summary>
    public Stream Payload { get; set; }

    /// <summary>
    ///  .
    ///         DLP-.
    /// </summary>
    public ContentType Type { get; set; }
}
```

Типичные модульные тесты на этот интерфейс могут выглядеть следующим образом:

```
    [TestFixture]
public sealed class DataLossPreventorTests
{
    private readonly IDataLossPreventor sut;

    ...

    [Test]
    public void When_Content_is_safe_Then_Rules_will_not_be_returned()
    {
        // Arrange
        var safeContent = new Content("Absolutely safe text", ContentType.Text);

        // Act
        var ruleSet = sut.Inspect("appID", safeContent );

        // Assert
        ruleSet.Should().BeEmpty();
    }

        [Test]
    public void When_ontent_contains_unsafe_data_a_collection_of_Rules_that_were_triggered_is_returned()
    {
        // Arrange
        var unsafeContent = new Content("Text with secret data such as #123-45-67", ContentType.Text);

        var expected = new Policy[]
        {
            new Policy("WORK-PHONE", "  ", Similarity.Likely, content.ContentID)
                               .AddLocation(new TextLocation(left: 30, right: 40))
        };

        // Act
        var actual = sut.Inspect("appID", unsafeContent);

        // Assert
        actual.Should().BeEquivalentTo(expected);
    }
}
```

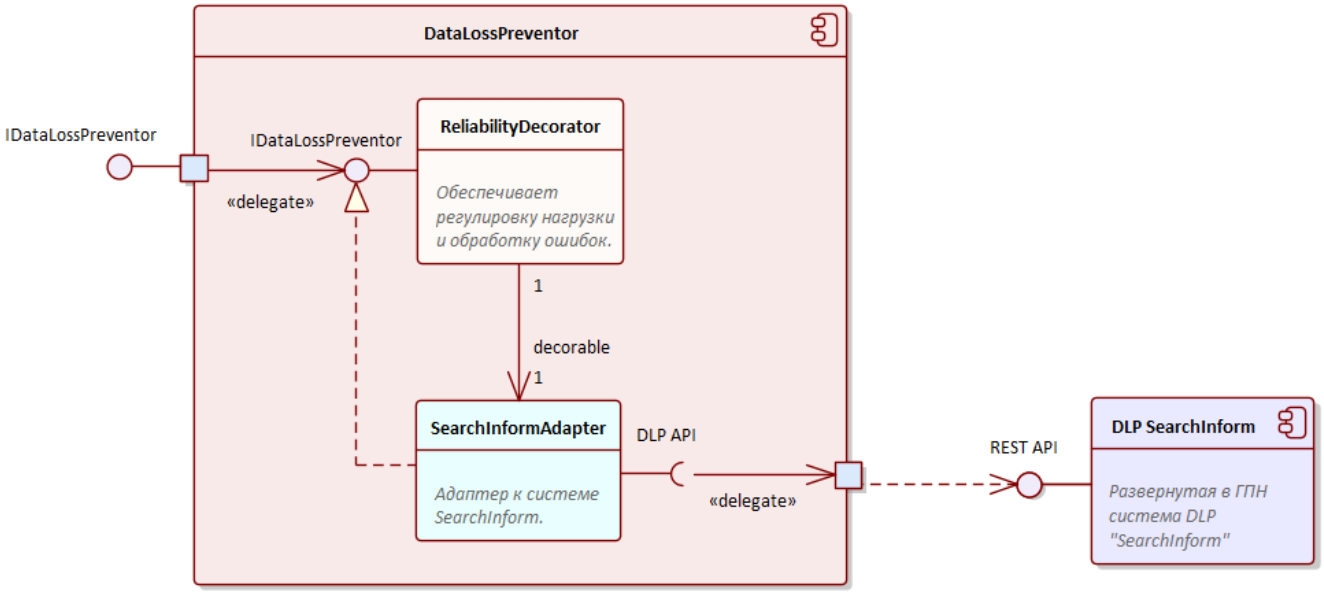
Очевидно, что второй тестовый метод разумнее сделать параметрическим.

## 4.2. Аналитическая модель ответа DLP-системы.




# 5. Предполагаемая реализация

Имеет смысл построить компонент DataLossPreventer как цепочку декораторов над адаптером реальной DLP-системы SearchInform (или другой, если эта будет заменена ГПН).
В декораторах удобно будет реализовать функционал обработки ошибок (а точнее - сокрытием ошибок реальной DLP-системы), восстановления после сбоев (паттерн `Retry`), регулировку нагрузки (то есть реализацию паттерна `CircuetBreaker`) и всего, что требуется для безопасной работы с внешними ресурсами.


DataLossPreventor
IDataLossPreventor
IDataLossPreventor
«delegate»
ReliabilityDecorator
Обеспечивает регулировку нагрузки и обработку ошибок.
1
decorable
1
SearchInformAdapter
Адаптер к системе SearchInform.
DLP API
«delegate»
REST API
DLP SearchInform
Развернутая в ГПН система DLP "SearchInform"