Лекции 5-6
«**Проектирование топологии информационных систем**»

Овчинников П.Е.
МГТУ «СТАНКИН»,
ст.преподаватель кафедры ИС

20.02.2019

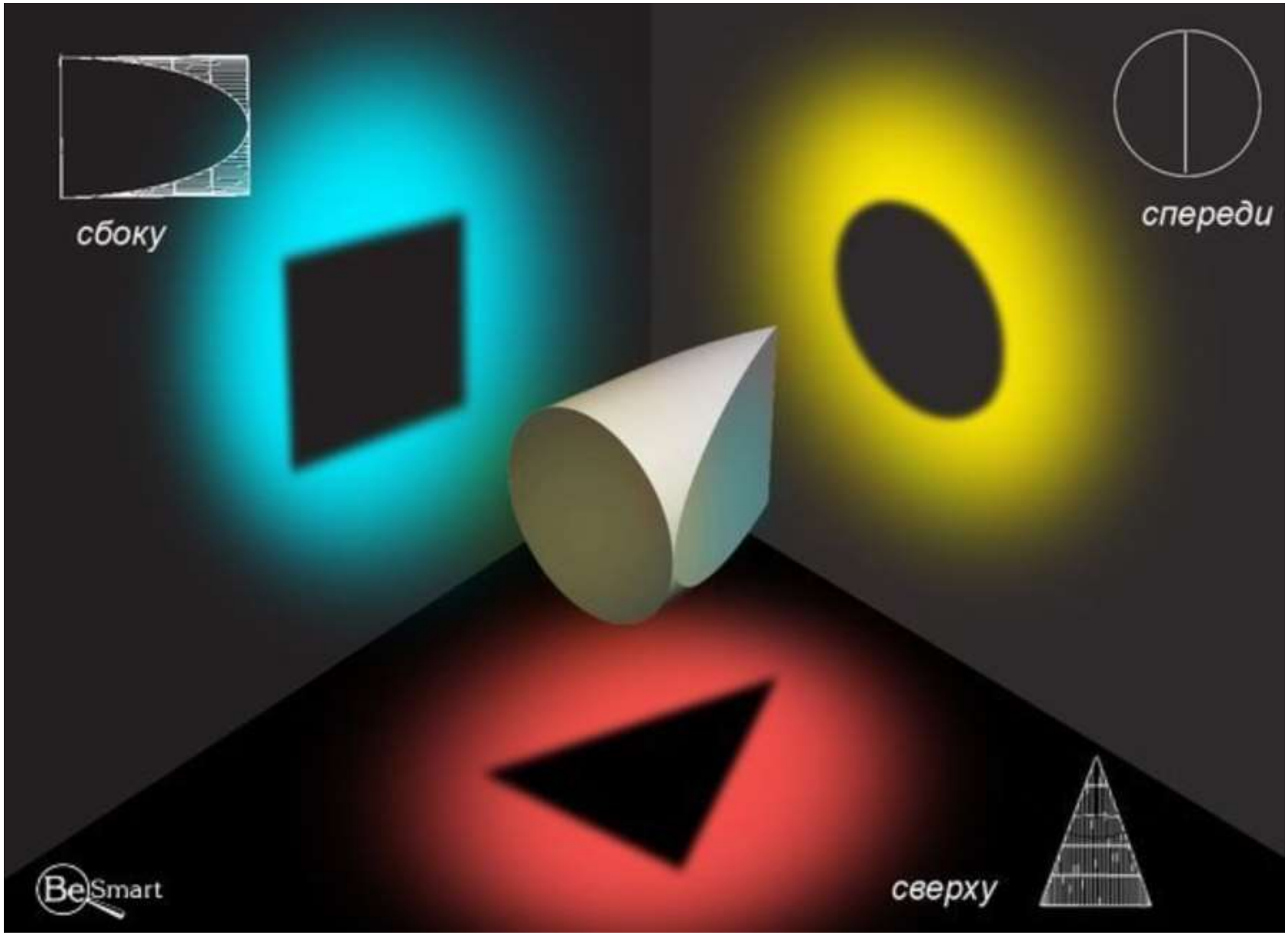
Дальше

Лекция 5
«Отличия понятий связности в применении к классам и объектам»

Овчинников П.Е.
МГТУ «СТАНКИН»,
ст.преподаватель кафедры ИС

13.02.2019

Точка зрения определяет все



Забавно

Топология в математике и сетях

Тополо́гия (от др.-греч. τόπος — место и λόγος — слово, учение) — раздел математики, изучающий:

- в самом общем виде — явление непрерывности;
- в частности — свойства пространств, которые остаются неизменными при непрерывных деформациях

В отличие от геометрии, в топологии не рассматриваются метрические свойства объектов (например, расстояние между парой точек). Например, с точки зрения топологии, кружка и бублик (полноторий) — неотличимы

Сетевая тополо́гия — это конфигурация графа, вершинам которого соответствуют конечные узлы сети (компьютеры) и коммуникационное оборудование (маршрутизаторы), а рёбрам — физические или информационные связи между вершинами

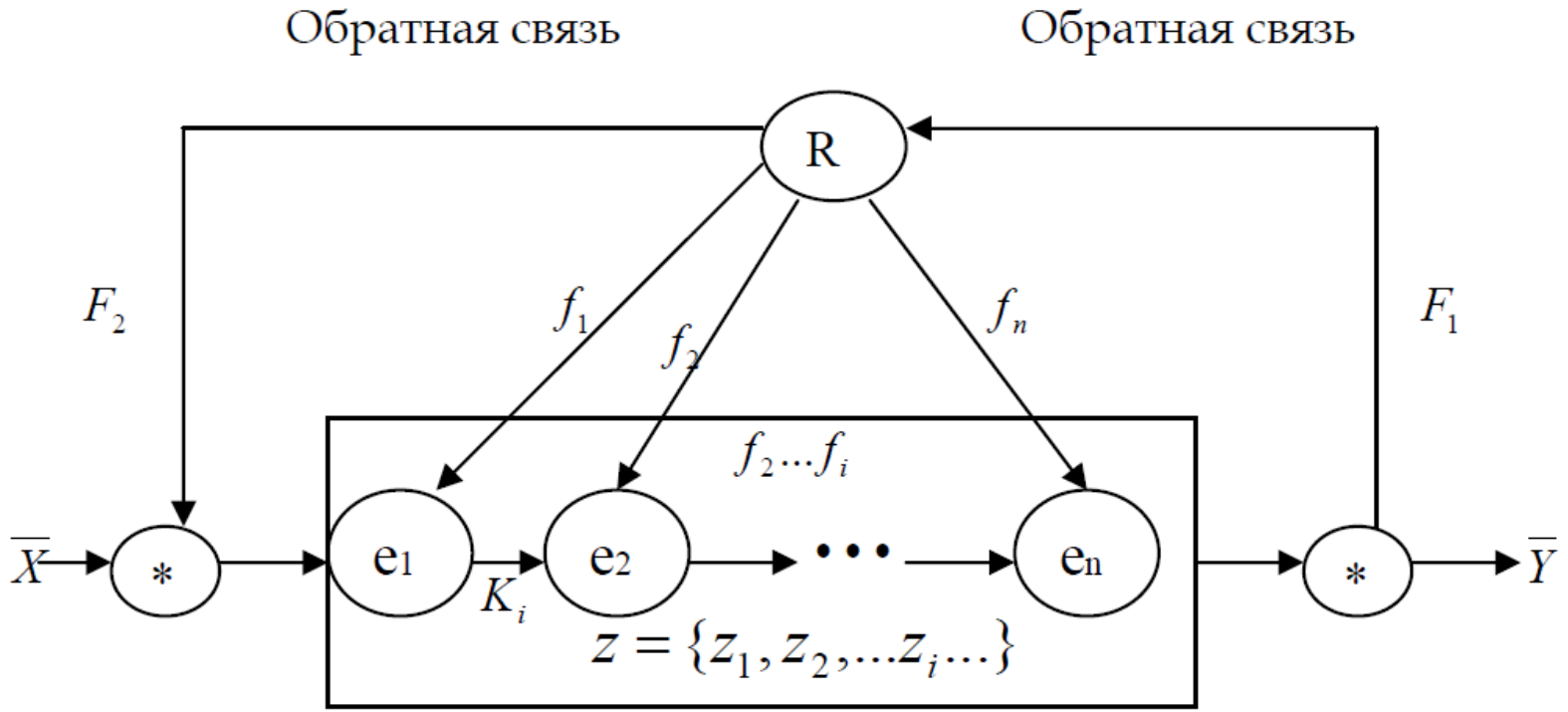
Сетевая топология может быть

- **физической** — описывает реальное расположение и связи между узлами сети
- **логической** — описывает хождение сигнала в рамках физической топологии
- **информационной** — описывает направление потоков информации, передаваемых по сети
- **управления обменом** — это принцип передачи права на пользование сетью

Топология (Википедия)

Топология в системах управления

Система ∑ – это конечная совокупность элементов (E) и некоторого регулирующего устройства (R), которое устанавливает связи между элементами (e_i) по преобразованию и управлению, управляет этими связями, создавая неделимую единицу функционирования. Топологически система представлена на рис. 1.

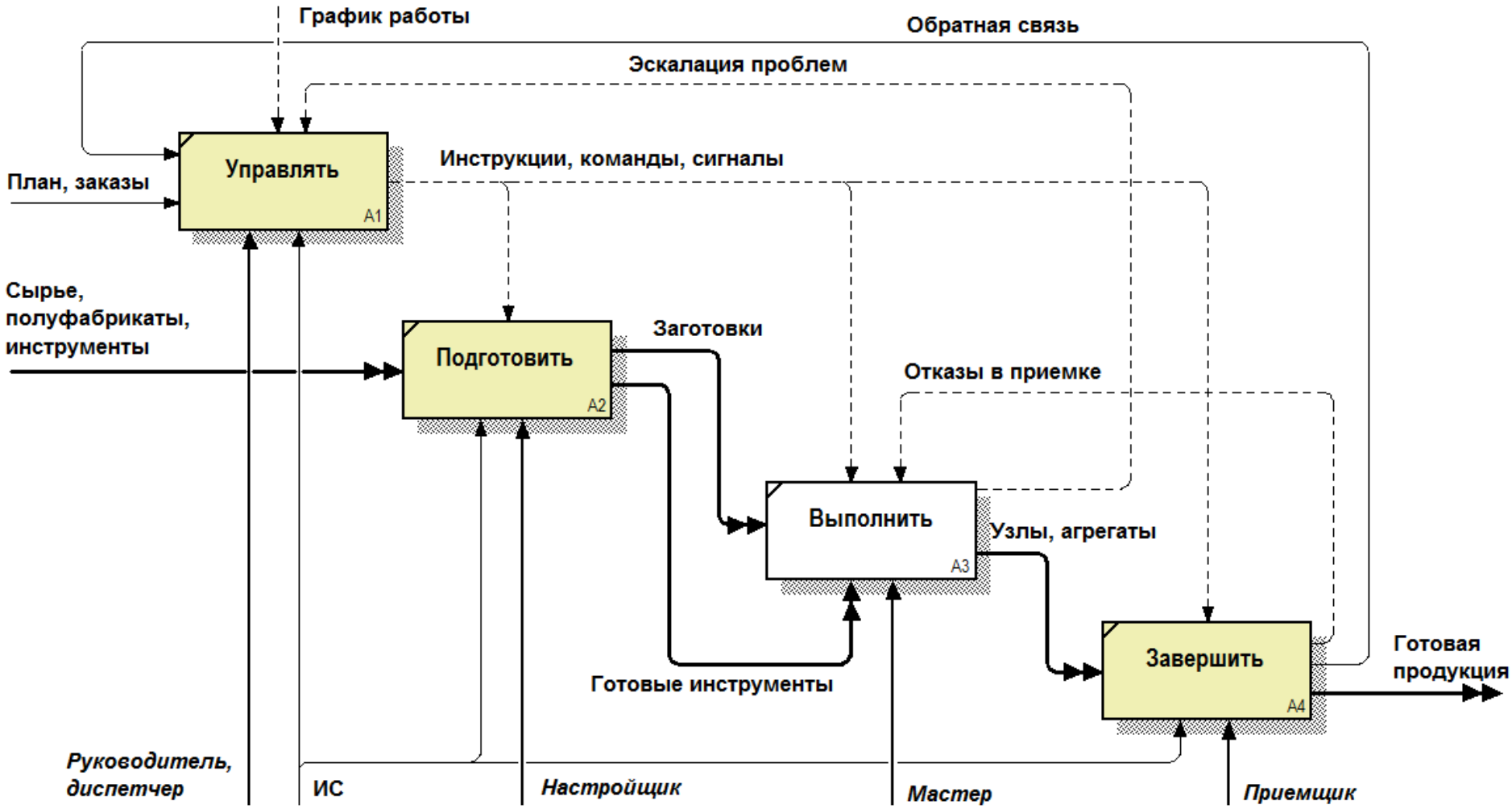


Данелян Т.Я.

Д177 ТЕОРИЯ СИСТЕМ И СИСТЕМНЫЙ АНАЛИЗ (ТСиСА): учебно-методический комплекс / Т.Я. Данелян. – М.: Изд. центр ЕАОИ, 2010. – 303 с.

ISBN 978-5-374-00324-6

Прямые и обратные связи в управлении



Классы и объекты

Класс (class) - категория вещей, которые имеют общие **атрибуты** и **операции**

Объект (object) -

- конкретная **материализация** абстракции;
- сущность с хорошо определенными границами, в которой **инкапсулированы состояние и поведение**;
- **экземпляр** класса

Объект **уникально идентифицируется** значениями атрибутов, определяющими его состояние в данный момент времени

Атрибут класса (class attribute) -
именованное свойство класса, описывающее множество значений, которые могут принимать экземпляры этого свойства

Операция класса (class operation) -
метод или функция, которая может быть выполнена экземпляром класса или интерфейсом

http://citforum.ru/database/advanced_intro/31.shtml
Свойство (property)

Объекты в программировании

Свойство объекта — способ доступа к внутреннему состоянию объекта, имитирующий переменную некоторого типа.

Метод объекта в объектно-ориентированном программировании — это функция или процедура, принадлежащая какому-то классу или объекту

Как и процедура в процедурном программировании, метод состоит из некоторого количества операторов для выполнения какого-то действия и имеет набор входных аргументов

Инкапсуляция (англ. *encapsulation*, от лат. *in capsula*) — в информатике упаковка данных и функций в единый компонент

В ООП инкапсуляция тесно связана с принципом абстракции данных (не путать с абстрактными типами данных, реализации которых предоставляют возможность инкапсуляции, но имеют иную природу). Это, в частности, приводит к другому распространённому заблуждению — рассмотрению инкапсуляции неотрывно от сокрытия. В частности, в сообществе C++ или Java принято рассматривать инкапсуляцию без сокрытия как неполноценную

Неявные классы: модуль, переменная, функция

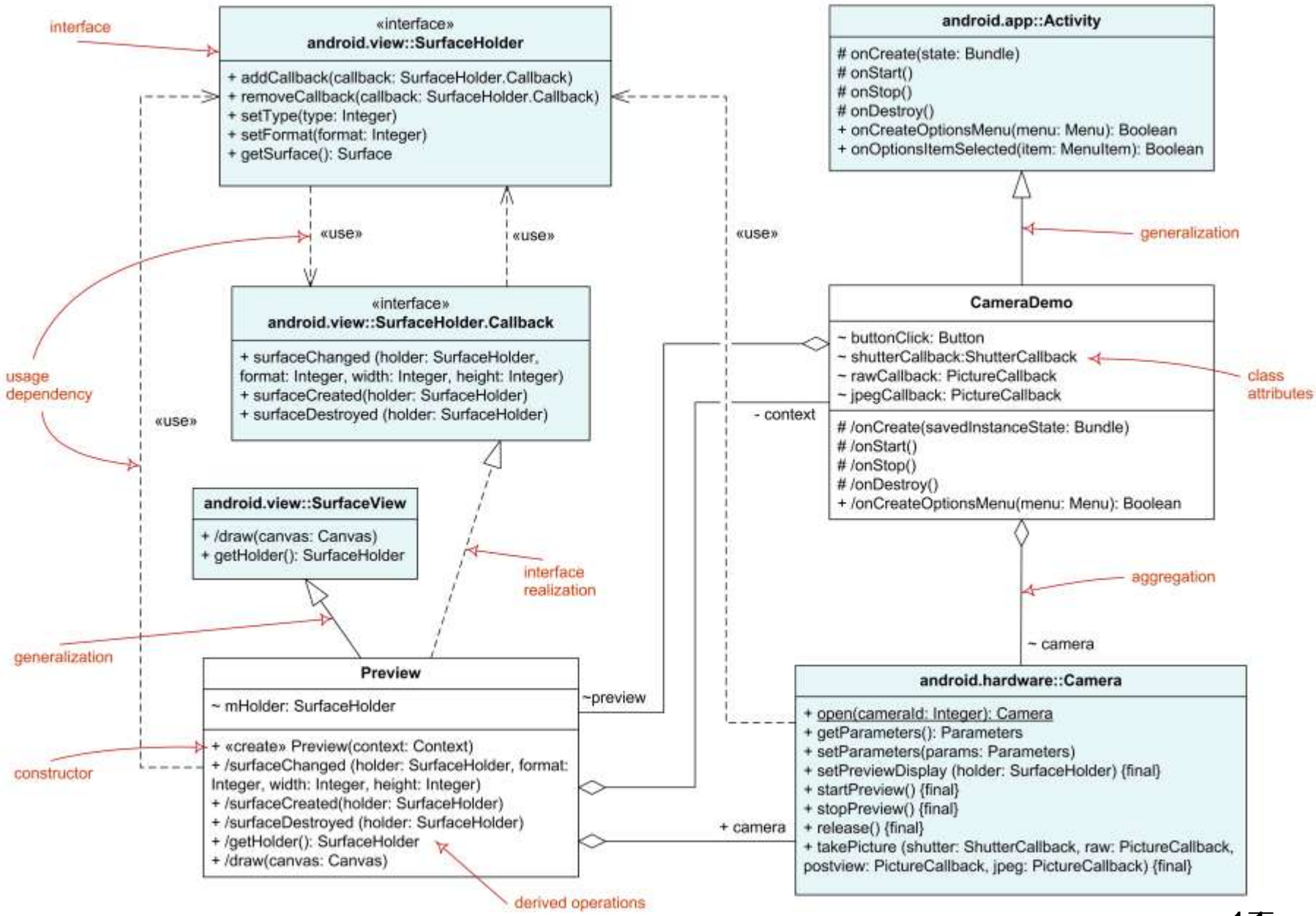
gauss12.js | iframe.html | uml-4.txt | uml-2.txt | task1-1.txt | uml-fdd.txt | new 1 | cat-1 rus.txt

```
1  var diagramColor = 'blueDiagram';
2  var diagramBorder = '';
3  var startTime;
4
5
6  /**
7   * Demonstrate Central limit theorem.
8   */
9  function count() {
10     startTime = new Date().getTime();
11
12     var n = parseInt(document.getElementById("number_of_sources").value) || 0;
13     var k = parseInt(document.getElementById("number_of_elements").value) || 0;
14
15     clearHistogram();
16     clearResult();
17
18     if(n === 0 || k === 0) {
19         alert("Пожалуйста введите данные!");
20         return;
21     }
22
23     var generatedNumbers = generate(n, k);
24
25     var array = generatedNumbers.array;
26     var min = generatedNumbers.min;
27     var expectedValue = generatedNumbers.expectedValue;
28     var deviation = generatedNumbers.deviation;
29     var gradY = generatedNumbers.gradY;
30
31
32     showHistogram(array, min, gradY);
33     showResult(k, array, expectedValue, deviation);
34 }
```

Неявные классы: объявления объектов

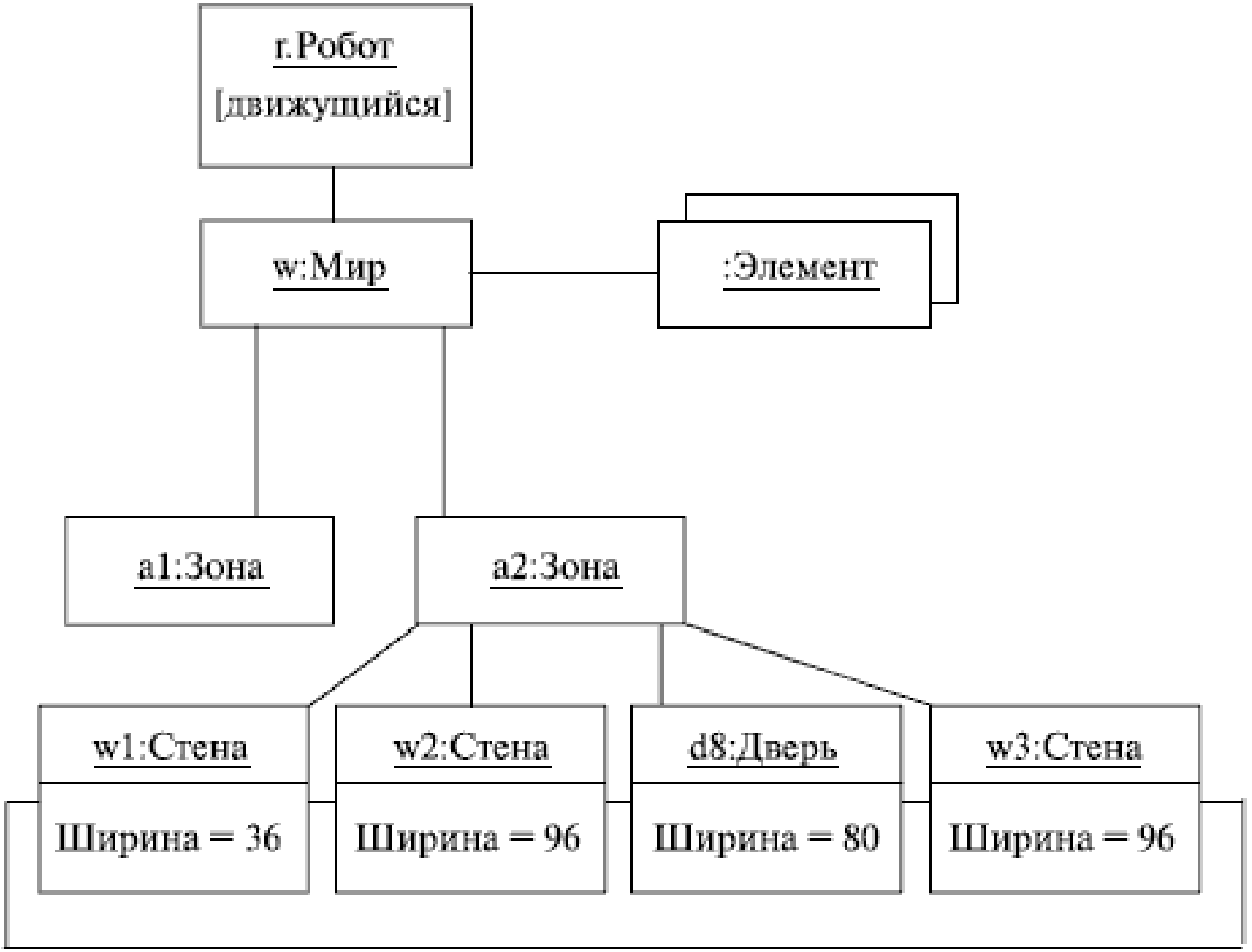
```
72      for(var i = 0; i < numberOfElements; i++) {
73          max = normalDistributedNumbers[i] > max ? normalDistributedNumb
74          min = normalDistributedNumbers[i] < min ? normalDistributedNumb
75      }
76      var gradY = (max - min) / 10;
77
78      var countList = [0,0,0,0,0,0,0,0,0,0];
79      for(var k = 0; k < numberOfElements; k++) {
80          var round = Math.floor(normalDistributedNumbers[k] / gradY - (m
81          countList[(round == 10) ? 9 : round] ++; // [1,2)-[2,3)-[3,4)-[
82      }
83
84      return {
85          array           : countList,
86          min             : min,
87          expectedValue   : expectedValue,
88          deviation       : deviation,
89          gradY           : gradY
90      };
91  }
```

Диаграммы классов



http://www.uml-diagrams.org/class-diagrams-overview.html

Диаграммы объектов (Object Diagram)



https://ru.wikipedia.org/wiki/UML
http://www.intuit.ru/studies/courses/1007/229/lecture/5950

Слой в многоуровневой архитектуре

В программной инженерии **многоуровневая архитектура** или **многослойная архитектура** — клиент-серверная архитектура, в которой разделяются функции:

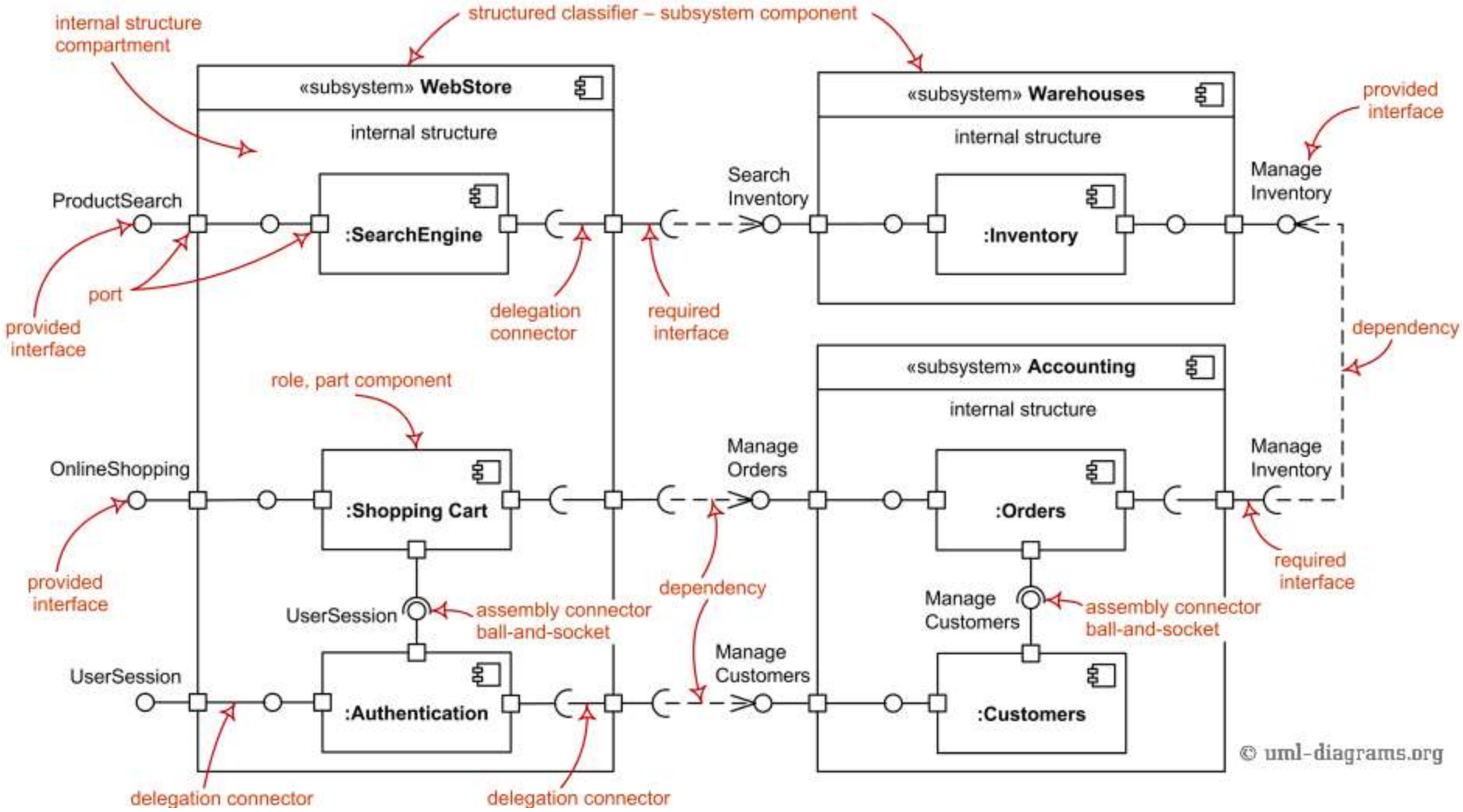
- представления,
- обработки и
- хранения данных.

N-уровневая архитектура приложения предоставляет модель, по которой разработчики могут создавать гибкие и повторно-используемые приложения.

Разделяя приложение на уровни абстракции, разработчики приобретают возможность внесения изменений в какой-то определённый слой, вместо того, чтобы перерабатывать всё приложение целиком.

Многоуровневая архитектура (Википедия)

Диаграмма компонентов



http://www.uml-diagrams.org/component-diagrams.html

Слои в многоуровневой архитектуре

Трёхуровневая архитектура обычно состоит из:

- уровня ***представления***,
- уровня ***бизнес логики*** и
- уровня ***хранения данных***

Хотя понятия слоя и уровня зачастую используются как взаимозаменяемые, многие сходятся во мнении, что между ними всё-таки есть различие:

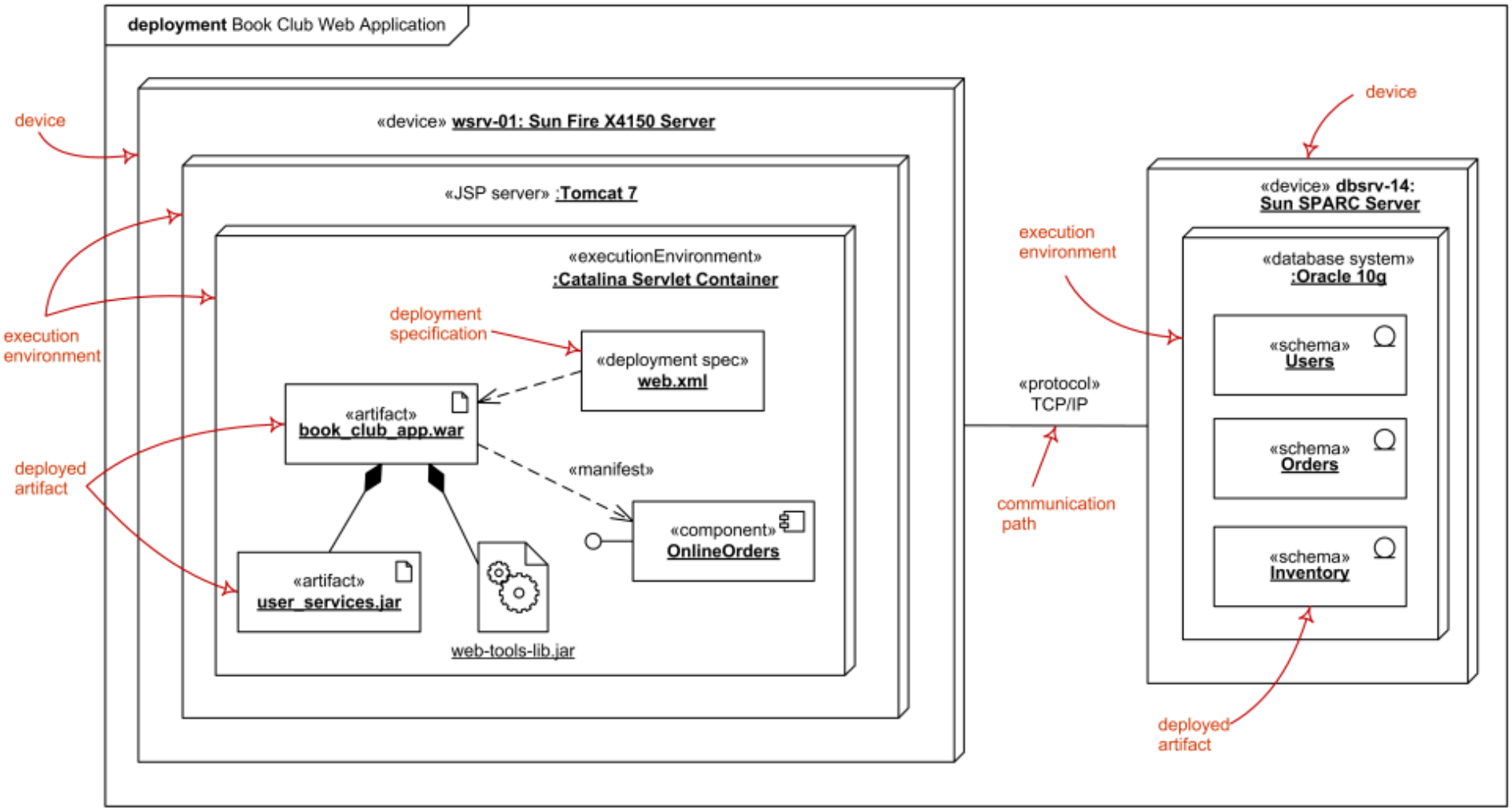
слой — это механизм **логического** структурирования компонентов, из которых состоит программное решение

уровень — это механизм **физического** структурирования инфраструктуры системы

Трёхуровневое решение легко может быть развёрнуто на единственном уровне, таком как персональная рабочая станция

Многоуровневая архитектура (Википедия)

Диаграмма развертывания

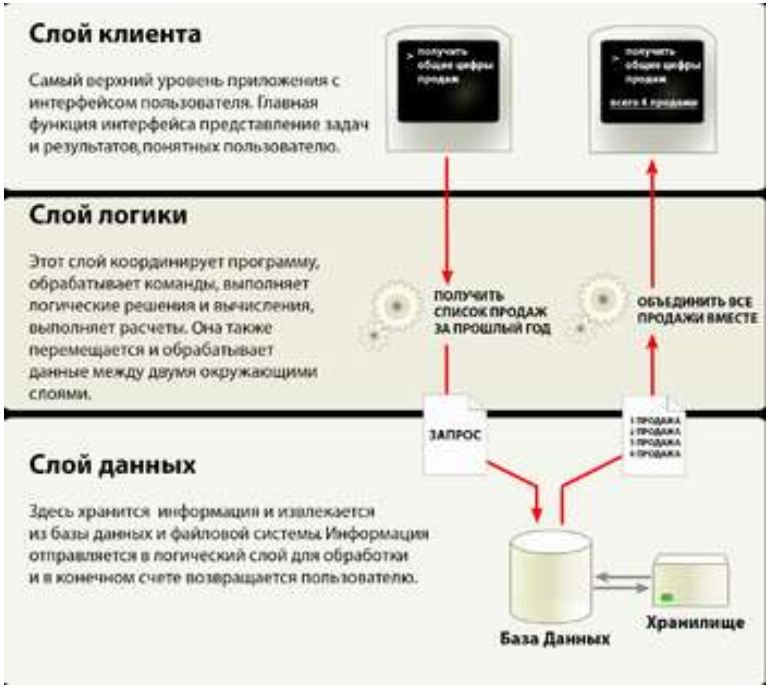


http://www.uml-diagrams.org/deployment-diagrams-overview.html

Многослойность

Клиент-сервер (англ. *Client-server*) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.

Трёху́ровневая архитекту́ра (*трёхзве́нная архитекту́ра*, англ. *three-layer*) — архитектурная модель программного комплекса, предполагающая наличие в нём трёх компонентов: клиента, сервера приложений (к которому подключено клиентское приложение) и сервера баз данных (с которым работает сервер приложений).баз.

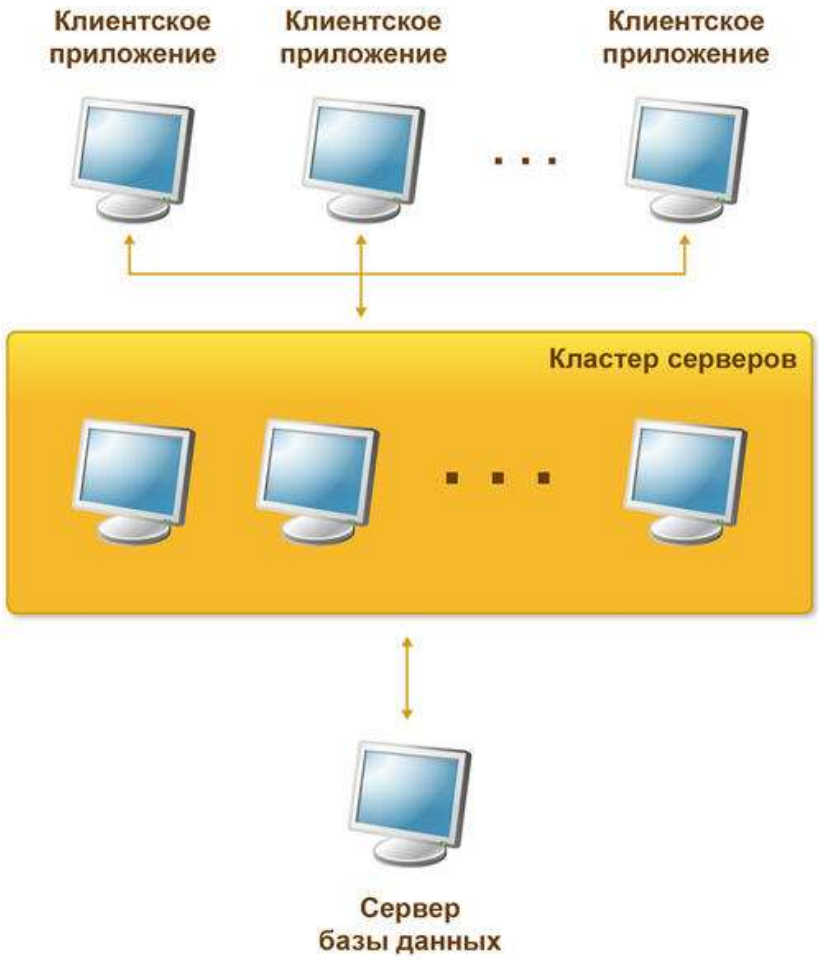


Клиент-сервер
Трехуровневая архитектура

Многослойность

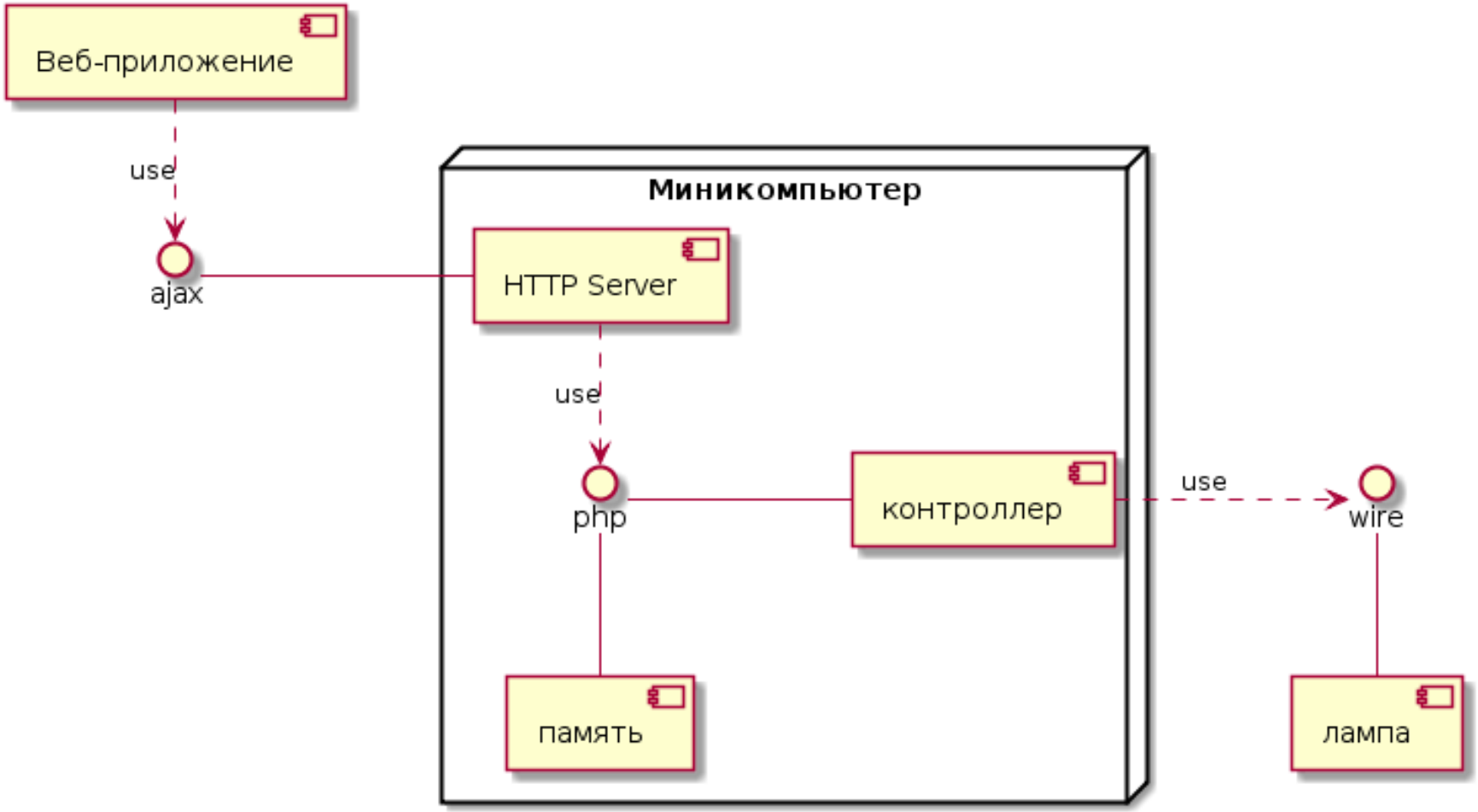
Кластер серверов 1С:Предприятия 8 - основной компонент платформы, обеспечивающий взаимодействие между пользователями и системой управления базами данных в клиент-серверном варианте работы. Наличие кластера позволяет обеспечить бесперебойную, отказоустойчивую, конкурентную работу большого количества пользователей с крупными информационными базами.

Кластер серверов 1С:Предприятия 8 является логическим понятием и представляет собой совокупность рабочих процессов, обслуживающих один и тот же набор информационных баз.



http://v8.1c.ru/overview/Term_000000126.htm

Простое web-приложение



Среда разработки и среда выполнения

Веб-приложение

use

ajax

Миникомпьютер

HTTP Server

use

php

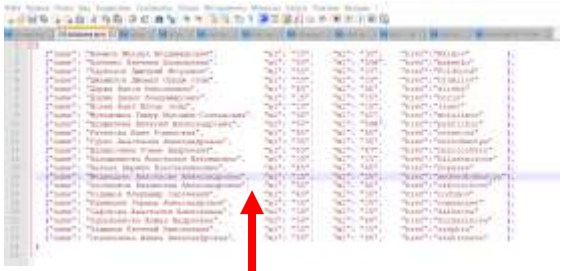
контроллер

use

wire

память

лампа



Deploy

Репозиторий проекта (Гитхаб)

Локальный репозиторий

Commit

Sync

Интернет: браузеры и URL

Бра́узер, или **веб-обозреватель** — прикладное программное обеспечение для просмотра веб-страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач.

HTML (от англ. *HyperText Markup Language* — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

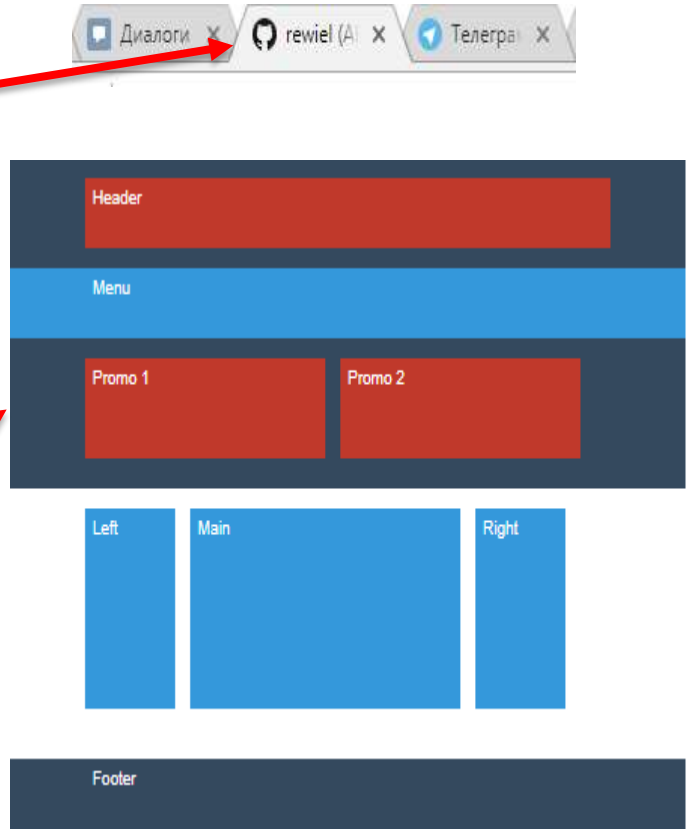
Единый указатель ресурса (англ. *Uniform Resource Locator*) — единообразный локатор (определитель местонахождения) ресурса.

традиционная форма записи URL:
<схема>:[//[<логин>:<пароль>@]<хост>[:<порт>]][/]<URL-путь>[?<параметры>][#<якорь>]

web-верстка

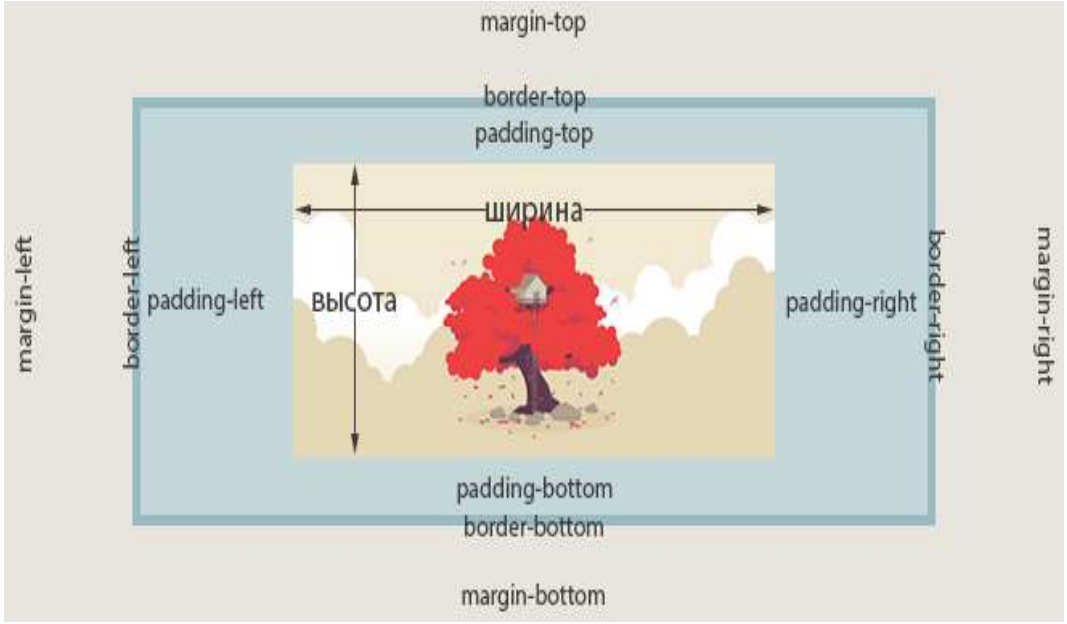
- <head> </head> - голова документа

  <title>название</title>
  <body> </body> - тело документа
- Все тэги, расположенные между <head> </head> что-то вроде служебной информации
- Все тэги, расположенные между <body> </body> - непосредственное содержание документа.



Блочные элементы

В блочной модели элемент рассматривается как **прямоугольный контейнер**, имеющий область содержимого, необязательные рамки и отступы (внутренние и внешние).



Блочные элементы — элементы высшего уровня, которые форматируются визуально как блоки, располагаясь на странице в окне браузера вертикально. Основные блочные элементы: **div, p, table, list-item**

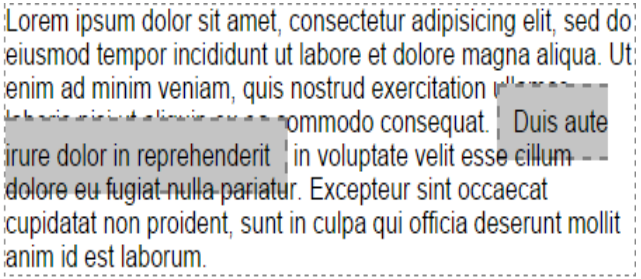
Строчные элементы

Встроенные (строчные) элементы генерируют внутристрочные контейнеры. Они не формируют новые блоки контента.

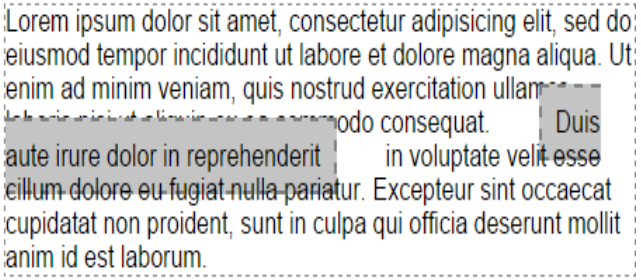
Строчные элементы являются **потомками блочных** элементов.

Ширина и высота строчного элемента зависит только от его содержимого

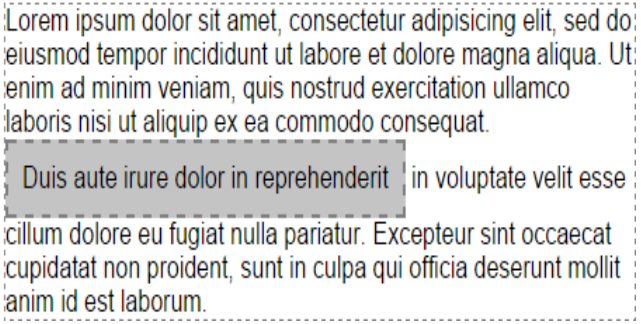
```
span {padding: 10px;
background: #c4c4c4;
border: 2px dashed grey}
```




```
span {padding: 10px;
margin: 30px;
background: #c4c4c4;
border: 2px dashed grey;}
```




```
span {display: inline-block;
padding: 10px;
background: #c4c4c4;
border: 2px dashed grey;}
```



Данные на клиенте: DOM

DOM (от англ. *Document Object Model* — «объектная модель документа») — это не зависящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получить доступ к содержимому HTML-, XHTML- и XML-документов, а также изменять **содержимое**, **структуру** и **оформление** таких документов

Модель DOM не накладывает ограничений на структуру документа

Любой документ известной структуры с помощью DOM может быть представлен в виде **дерева узлов**, каждый узел которого представляет собой:

- **элемент**
- **атрибут**
- текстовый, графический или **любой другой объект**.

Узлы связаны между собой отношениями «родительский-дочерний»

Интернет: интерактивность на клиенте

Dynamic HTML или **DHTML** — это способ (подход) создания **интерактивного** веб-сайта, использующий сочетание

- статичного языка разметки HTML,
- встраиваемого и выполняемого на стороне клиента скриптового языка JavaScript
- CSS (каскадных таблиц стилей) и
- DOM (объектной модели документа)

Он может быть использован для создания приложения в веб-браузере, например, для более простой навигации или для придания интерактивности форм

DHTML может быть использован для динамического перетаскивания элементов по экрану

Также он может служить как инструмент для создания основанных на браузере графических приложений и видеоигр

Веб-разработка: AJAX

AJAX базируется на использовании технологий:

- динамического обращения к серверу «на лету», без перезагрузки всей страницы полностью, например с использованием XMLHttpRequest и
- использования DHTML для динамического изменения содержания страницы

XMLHttpRequest (XMLHTTP, XHR) — API, доступный в скриптовых языках браузеров, таких как JavaScript

Использует запросы HTTP или HTTPS напрямую к веб-серверу и загружает данные ответа сервера напрямую в вызывающий скрипт

Информация может передаваться в любом текстовом формате, например, в XML, HTML или JSON. Позволяет осуществлять HTTP-запросы к серверу без перезагрузки страницы

HTMLHttpRequest (Википедия)

Взаимодействие с http-сервером: AJAX

```
<!DOCTYPE html>
<html>
<head>
<script>
function showHint(str) {
    if (str.length == 0) {
        document.getElementById("txtHint").innerHTML = "";
        return;
    } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
            //alert(this.readyState+' '+this.status);
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById("txtHint").innerHTML = this.responseText;
            }
        }
        xmlhttp.open("GET", "dummy.php?q="+str, true);
        xmlhttp.send();
    }
}
</script>
</head>
<body>

<p><b>Введите данные в строку:</b></p>
<form>
Строка: <input type="text" onkeyup="showHint(this.value)">
</form>
<p>Ответ:<br><span id="txtHint"></span></p>
</body>
</html>
```

Взаимодействие с http-сервером: AJAX

```php
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $parm = $_POST;
    echo('Метод POST ('.count($parm).'):<br>');
} else {
    $parm = $_GET;
    echo('Метод GET ('.count($parm).'):<br>');
}
foreach($parm as $k=>$v) {
    echo('<b>'.$k.'</b> = '.trim(stripslashes(htmlspecialchars($v))).'<br>');
}
?>
```

Взаимодействие с http-сервером: AJAX

jQuery ajax() Method

‹ jQuery AJAX Methods

Example

Change the text of a element using an AJAX request:

```
$("button").click(function(){
  $.ajax({url: "demo_test.txt", success: function(result){
    $("#div1").html(result);
  }});
});
```

Try it Yourself »

JQuery AJAX

Взаимодействие с http-сервером: cURL

cURL — (распространяемая по лицензии MIT), кроссплатформенная служебная программа командной строки, позволяющая взаимодействовать с множеством различных серверов по множеству различных протоколов с синтаксисом URL

Программа cURL может автоматизировать передачу файлов или последовательность таких операций. Например, это хорошее средство для **моделирования действий пользователя** в веб-обозревателе.

Программа поддерживает протоколы:
FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, LDAP, а также POP3, IMAP и SMTP.

Также cURL поддерживает сертификаты HTTPS, методы HTTP POST, HTTP PUT, загрузку на FTP, загрузку через формы HTTP.

Поддерживаемые методы аутентификации: базовая, дайджест, NTLM и Negotiate для HTTP, а также Kerberosдля FTP.

Взаимодействие с http-сервером: WebSocket

WebSocket — протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени (online)

В настоящее время в W3C осуществляется стандартизация API Web Sockets. Черновой вариант стандарта этого протокола утверждён IETF

WebSocket разработан для воплощения в веб-браузерах и веб-серверах, но он может быть использован для любого клиентского или серверного приложения.

Протокол WebSocket — это независимый протокол, основанный на протоколе TCP. Он делает возможным более тесное взаимодействие между браузером и веб-сайтом, способствуя распространению интерактивного содержимого и созданию приложений реального времени

Распределенные вычисления

Распределённые вычисления — способ решения трудоёмких вычислительных задач с использованием нескольких компьютеров, чаще всего объединённых в параллельную вычислительную систему

Грид-вычисления (англ. *grid* — решётка, сеть) — это форма распределённых вычислений, в которой «виртуальный суперкомпьютер» представлен в виде кластеров, соединённых с помощью сети, слабосвязанных гетерогенных компьютеров, работающих вместе для выполнения огромного количества заданий (операций, работ). Эта технология применяется для решения научных, математических задач, требующих значительных вычислительных ресурсов. Грид-вычисления используются также в коммерческой инфраструктуре для решения таких трудоёмких задач, как экономическое прогнозирование, сейсмоанализ, разработка и изучение свойств новых лекарств.

BOINC (англ. ***Berkeley Open Infrastructure for Network Computing***) — открытая программная платформа (университета Беркли для GRID вычислений) — некоммерческое межплатформенное ПО для организации распределённых вычислений. Используется для организации добровольных вычислений

Параллельные алгоритмы

Параллельные вычисления — способ организации компьютерных вычислений, при котором программы разрабатываются как набор взаимодействующих вычислительных процессов, работающих параллельно (одновременно)

Термин охватывает совокупность вопросов параллелизма в программировании, а также создание эффективно действующих аппаратных реализаций

Теория параллельных вычислений составляет раздел прикладной теории алгоритмов

Параллельные программы могут физически исполняться либо последовательно на единственном процессоре — перемежая по очереди шаги выполнения каждого вычислительного процесса, либо параллельно — выделяя каждому вычислительному процессу один или несколько процессоров (находящихся рядом или распределённых в компьютерную сеть).

Основная сложность при проектировании параллельных программ — обеспечить правильную последовательность взаимодействий между различными вычислительными процессами, а также **координацию** ресурсов, разделяемых между процессами

Параллельные вычисления (Википедия)

Лекция 6
«Локальные, удаленные и распределенные базы данных»

Овчинников П.Е.
МГТУ «СТАНКИН»,
ст.преподаватель кафедры ИС

20.02.2019

Интернет: базы данных на клиенте

Web SQL (или **Web SQL Database**) — это API веб-страниц для хранения данных в веб-браузере на основе SQL

API поддерживается Google Chrome, Opera, Safari и браузером Android

Консорциум W3C прекратил работу над спецификацией в ноябре 2010 года, в качестве причины завершения спецификации ссылаясь на отсутствие независимых реализаций (т.е. систем баз данных отличных от SQLite в качестве внутреннего интерфейса), из-за чего спецификации этого API не входит в список рекомандованых W3C

IndexedDB — JavaScript интерфейс прикладного программирования (API) клиентского хранилища большого объема структурированных данных, в том числе файлы/blobs

Другими словами это NoSQL хранилище данные в формате JSON внутри браузера

Стандарт разработан W3C и внедрен в браузерах с 2011 года

Интернет: веб-сервер

Веб-сервер — сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными.

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер (см.: Сервер (аппаратное обеспечение)), на котором это программное обеспечение работает.

Веб-серверы могут иметь различные дополнительные функции, например:

- автоматизация работы веб-страниц;
- ведение журнала обращений пользователей к ресурсам;
- аутентификация и авторизация пользователей;
- поддержка динамически генерируемых страниц;
- поддержка HTTPS для защищённых соединений с клиентами.

В терминологии компьютерных сетей, **балансировка нагрузки**, или **выравнивание нагрузки** (англ. *load balancing*) — метод распределения заданий между несколькими сетевыми устройствами (например, серверами) с целью оптимизации использования ресурсов, сокращения времени обслуживания запросов

Интернет: сервер приложений

Сервер приложений (англ. *application server*) — это программная платформа (фреймворк), предназначенная для эффективного исполнения процедур (программ, скриптов), на которых построены приложения.

Сервер приложений действует как набор компонентов, доступных разработчику программного обеспечения через API (интерфейс прикладного программирования), определённый самой платформой.

Для веб-приложений основная задача компонентов сервера — обеспечивать создание динамических страниц. Однако современные серверы приложений включают в себя и поддержку кластеризации, повышенную отказоустойчивость, балансировку нагрузки, позволяя таким образом разработчикам сфокусироваться только на реализации бизнес-логики

В случае Java-сервера приложений, сервер приложений ведёт себя как расширенная виртуальная машина для запуска приложений, прозрачно управляя соединениями с базой данных, с одной стороны, и соединениями с веб-клиентом, с другой

Интернет: веб-служба

Веб-служба, *веб-сервис* (англ. *web service*) — идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя

Веб-службы могут взаимодействовать друг с другом и со сторонними приложениями посредством сообщений, основанных на определённых протоколах (SOAP, XML-RPC и т. д.) и соглашениях (REST)

Веб-служба является единицей модульности при использовании сервис-ориентированной архитектуры приложения

В обиходе *веб-сервисами* называют услуги, оказываемые в Интернете. В этом употреблении термин требует уточнения, идёт ли речь о поиске, веб-почте, хранении документов, файлов, закладок и т. п. Такими веб-сервисами можно пользоваться независимо от компьютера, браузера или места доступа в Интернет

Интернет: сервер баз данных

Сервер баз данных (БД) выполняет обслуживание и управление базой данных и отвечает за целостность и сохранность данных, а также обеспечивает операции ввода-вывода при доступе клиента к информации

Архитектура клиент-сервер состоит из клиентов и серверов. Основная идея состоит в том, чтобы размещать серверы на мощных машинах, а приложениям, использующим языковые компоненты СУБД, обеспечить доступ к ним с менее мощных машин-клиентов посредством внешних интерфейсов

Систе́ма управле́ния ба́зами да́нных, *сокр.* СУБД (англ. *Database Management System*, сокр. DBMS) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных

СУБД — комплекс программ, позволяющих создать базу данных (БД) и манипулировать данными (вставлять, обновлять, удалять и выбирать)

Система обеспечивает безопасность, надёжность хранения и целостность данных, а также предоставляет средства для администрирования БД

Транзакции (информатика)

Транза́кция (англ. *transaction*) — группа последовательных операций с базой данных, которая представляет собой логическую единицу работы с данными

Транзакция может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от параллельно идущих других транзакций, либо не выполнена вообще, и тогда она не должна произвести никакого эффекта

Транзакции обрабатываются транзакционными системами, в процессе работы которых создаётся история транзакций

Различают последовательные (обычные), параллельные и распределённые транзакции. Распределённые транзакции подразумевают использование более чем одной транзакционной системы и требуют намного более сложной логики (например, two-phase commit — двухфазный протокол фиксации транзакции)

Также в некоторых системах реализованы автономные транзакции, или подтранзакции, которые являются автономной частью родительской транзакции

Транзакция (информатика)

Распределенные транзакции

Задача двух генералов — в вычислительной технике мысленный эксперимент, призванный проиллюстрировать проблему синхронизации состояния двух систем по ненадёжному каналу связи.

Откат (англ. rollback)
Системы обработки транзакций обеспечивают целостность базы данных при помощи записи промежуточного состояния базы данных перед её изменением, а затем, используя эти записи, восстанавливают базу данных до известного состояния, если транзакция не может быть совершена.

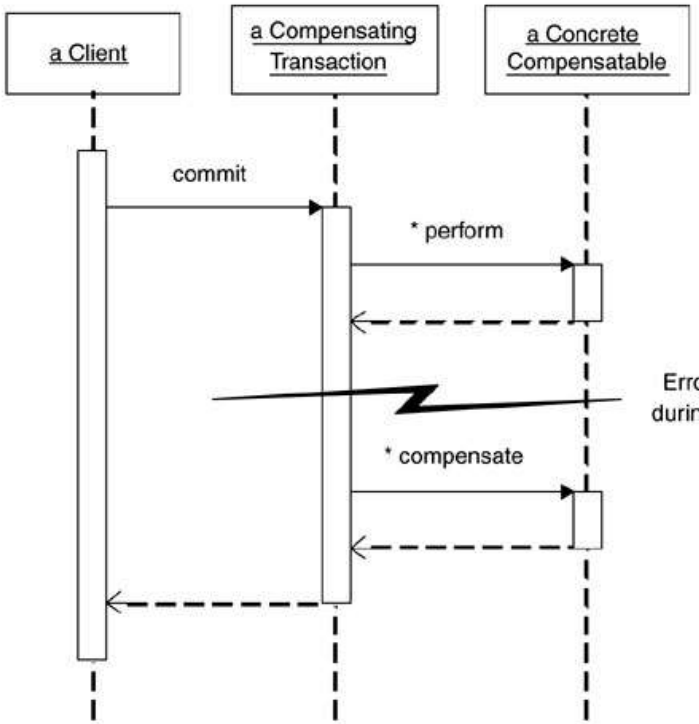
Прогон (англ. rollforward)
Кроме того, можно вести отдельный журнал всех изменений базы данных (иногда это называется after images); это не требует отката неудачных операций, но это полезно для обновления базы данных в случае отказа базы данных, поэтому некоторые системы обработки транзакций обеспечивают эту функцию.

Взаимная блокировка (англ. deadlocks)
В некоторых случаях, две транзакции могут в ходе их обработки пытаться получить доступ к одной и той же части базы данных в одно и то же время, таким образом, что это будет препятствовать их совершению. Системы обработки транзакций предназначены для обнаружения таких ситуаций. Обычно обе транзакции отменяются и производится откат, а затем они автоматически запускаются в другом порядке, так что взаимоблокировка не повторится.

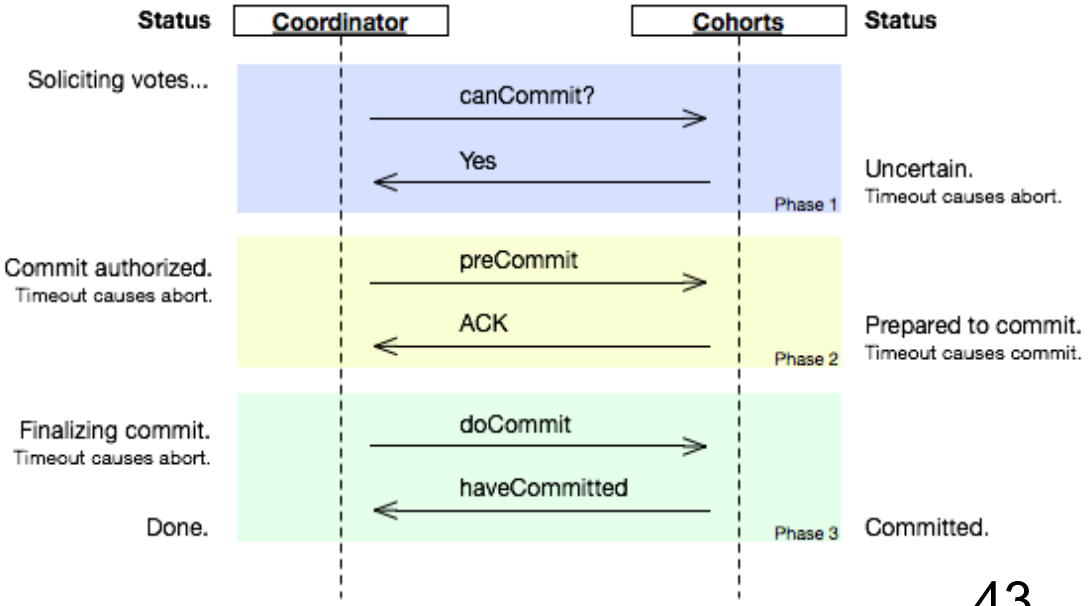
Задача двух генералов

Многофазная обработка транзакций

Отмена (abort) и компенсация (compensation)



Двухфазная обработка



Трехфазная обработка

Распределенные реестры

Технология распределенных реестров (Distributed ledger technology, DLT) — это технология хранения информации, ключевыми особенностями которой являются:

- совместное использование и синхронизация цифровых данных согласно алгоритму консенсуса
- географическое распределение равнозначных копий в разных точках по всему миру
- отсутствие центрального администратора

Распределенные реестры

Ключевая особенность распределенного реестра — отсутствие единого центра управления.

Каждый узел составляет и записывает обновления реестра независимо от других узлов. Затем узлы голосуют за обновления, чтобы удостовериться, что большинство узлов согласно с окончательным вариантом.

Голосование и достижение согласия в отношении одной из копий реестра называется консенсусом, этот процесс выполняется автоматически с помощью алгоритма консенсуса.

Как только консенсус достигнут, распределенный реестр обновляется, и последняя согласованная версия реестра сохраняется в каждом узле.

Распределенные реестры

Задача двух генералов — в вычислительной технике мысленный эксперимент, призванный проиллюстрировать проблему синхронизации состояния двух систем по ненадёжному каналу связи, в литературе также иногда упоминается как **задача двух армий**.

Определение: две армии, каждая руководимая своим генералом, готовятся к штурму города. Лагеря этих армий располагаются на двух холмах, разделённых долиной. Единственным способом связи между генералами является отправка посыльных с сообщениями через долину. Но долина занята противником и любой из посыльных может быть перехвачен. Проблема заключается в том, что, генералы заранее (пока была связь) приняли принципиальное решение о штурме, но не согласовали точное время штурма.

Распределенные реестры

Теорема CAP (известная также как **теорема Брюера**) — эвристическое утверждение о том, что в любой реализации распределённых вычислений возможно обеспечить не более двух из трёх следующих свойств:

- *согласованность данных* (англ. *consistency*) — во всех вычислительных узлах в один момент времени данные не противоречат друг другу;
- *доступность* (англ. *availability*) — любой запрос к распределённой системе завершается корректным откликом, однако без гарантии, что ответы всех узлов системы совпадают;
- *устойчивость к разделению* (англ. *partition tolerance*) — расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций.

Распределенные реестры

Задача византийских генералов (англ. *Byzantine fault tolerance (BFT), Byzantine agreement problem, Byzantine generals problem, Byzantine failure*) — в криптологии задача взаимодействия нескольких удаленных абонентов, которые получили приказы из одного центра. Часть абонентов, включая центр, могут быть злоумышленниками.

Формулировка: N «белых» генералов возглавляют армии в горах и готовятся атаковать «черных» в долине. Для связи атакующие используют надёжную связь (например, телефон). Однако из N генералов M являются предателями и активно пытаются воспрепятствовать согласию лояльных генералов. Согласие состоит в том, чтобы все лояльные генералы узнали о численности всех лояльных армий и пришли к одинаковым выводам (пусть и ложным) относительно состояния предательских армий.

Блокчейн

Blockchain — это один из видов **распределенного реестра**.

Они используют независимые компьютеры (ноды, узлы) для записи, совместного использования и синхронизации транзакций в своих соответствующих электронных реестрах.

Блокчейн организует данные в блоки, которые соединены вместе в режиме «только добавление».

- блок не может конфликтовать с другими данными, которые уже находятся в базе данных (непротиворечивость),
- это система Append-Only (неизменяемость — нельзя изменить какую-то часть внутри цепи (один блок), потому что другие блоки содержат информацию о нем),
- сами данные формируются их владельцем (контроль за собственной информацией — у создателя транзакции есть приватные ключи, их никто не знает),
- блокчейн можно копировать и просматривать некоторую информацию (прозрачность)

PKI

Инфраструктура открытых ключей (ИОК, англ. ***PKI*** *- Public Key Infrastructure*) — набор средств (технических, материальных, людских и т. д.), распределённых служб и компонентов, в совокупности используемых для поддержки криптозадач на основе закрытого и открытого ключей. В основе PKI лежит использование криптографической системы с открытым ключом и несколько основных принципов:

- закрытый ключ (private key) известен только его владельцу
- удостоверяющий центр создает электронный документ — сертификат открытого ключа, таким образом удостоверяя факт того, что закрытый (секретный) ключ известен эксклюзивно владельцу этого сертификата, открытый ключ (public key) свободно передается в сертификате
- никто не доверяет друг другу, но все доверяют удостоверяющему центру
- удостоверяющий центр подтверждает или опровергает принадлежность открытого ключа заданному лицу, которое владеет соответствующим закрытым ключом.

Майнинг

Майнинг, также **добыча** (от англ. *mining — добыча полезных ископаемых*) — деятельность по созданию новых структур (обычно речь идёт о новых блоках в блокчейне) для обеспечения функционирования криптовалютных платформ.

За создание очередной структурной единицы обычно предусмотрено вознаграждение за счёт новых (эмитированных) единиц криптовалюты и/или комиссионных сборов. Обычно майнинг сводится к серии вычислений с перебором параметров для нахождения хеша с заданными свойствами.

Разные криптовалюты используют разные модели вычислений, но они всегда достаточно длительны по времени для нахождения приемлемого варианта и быстры для проверки найденного решения (см. Доказательство выполнения работы). Такие вычисления используются алгоритмами криптовалют для обеспечения защиты от повторного расходования одних и тех же единиц, а вознаграждение стимулирует людей расходовать свои вычислительные мощности и поддерживать работу сетей.

ICO

ICO, ***Initial coin offering***, (с англ. — «первичное предложение монет, первичное размещение монет») — форма привлечения инвестиций в виде продажи инвесторам фиксированного количества новых единиц криптовалют, полученных разовой или ускоренной эмиссией.

Встречается также форма «первичного предложения токенов». Помимо этого термин ICO часто заменяется словом «краудсейл»

Токен — это единица учёта, не являющаяся криптовалютой, предназначенная для представления цифрового баланса в некотором активе, иными словами выполняющая функцию «заменителя ценных бумаг» в цифровом мире.

Токены представляют собой запись в регистре, распределенную в блокчейн-цепочке. Получить доступ к токену можно через специальные приложения, которые используют схемы электронной подписи. Основная часть существующих на сегодняшний день токенов формируется на протоколе Blockchain от Ethereum

Смарт-контракт

Смарт-контракт (англ. *Smart contract* — умный контракт) — компьютерный алгоритм, предназначенный для заключения и поддержания коммерческих контрактов в технологии блокчейн

Стороны подписывают умный контракт, используя методы, аналогичные подписанию отправки средств в действующих криптовалютных сетях. После подписания сторонами контракт вступает в силу. Для обеспечения автоматизированного исполнения обязательств контракта непременно требуется среда существования, которая позволяет полностью автоматизировать выполнение пунктов контракта. Это означает, что умные контракты смогут существовать только внутри среды, имеющей беспрепятственный доступ исполняемого кода к объектам умного контракта. Все условия контракта должны иметь математическое описание и ясную логику исполнения.

IOTA

IOTA расшифровывается как Internet of Things Application. Это новая криптотехнология, которая облегчает транзакции между устройствами в Интернете вещей (IoT)

IOTA исследует транзакционные сборы и проблемы масштабируемости технологий блокчейн. Проект избавился от блоков и их цепи. Вместо этого, чтобы отправить транзакцию в публичный реестр IOTA, вы должны проверить две другие предыдущие транзакции. Этот метод проверки означает, что нет центрального реестра и нет необходимости в том, чтобы майнеры поддерживали сеть.

Поскольку вычислительная мощность в Tangle растет по мере роста сети, IOTA обещает бесплатные быстрые транзакции. Проект предназначен для обработки микроплатежей и платежей между устройствами, что облегчает создание микроэкономики в машиностроении.