

**СЕМАНТИЧЕСКИЕ И СЕРВИСНЫЕ РЕСУРСЫ
ДЛЯ РАЗРАБОТКИ ОНТОЛОГИИ НАУЧНОЙ И
ИНЖЕНЕРНОЙ ПРЕДМЕТНЫХ ОБЛАСТЕЙ И ВЕБ-
СИСТЕМ**

Конференции

Scinces and Information-2015 London

"Научный сервис в сети Интернет"

19-24 сентября 2016, Абрау

"Научный сервис в сети Интернет"

19-24 сентября 2018, Абрау

Докладчик

**Доктор физ.- мат.наук, профессор МФТИ
Лаврищева Е.М.**

ТЕМАТИКА РАЗДЕЛОВ ДОКЛАДА

Общая характеристика современной электронной науки E-Science и развития информатики и искусственного интеллекта

1. Семантические и сервисные средства для разработки прикладных предметных областей знаний и Веб-систем

2. Разработка онтологии Вычислительной геометрии и домена Жизненного цикла стандарта ISO/IEC 12207-2007, 2012.

3. Веб-сервисы Интернет и методы создания современных Веб-систем и сайтов в Интернет

Электронная наука (e-Science, eScience) (1999)
e-Инфраструктура (e-Infrastructure, в Европе)
e-Киберинфраструктура (e-Cyberinfrastructure, в США)
e-Исследования (e-research) (2004)
(включая e-социальную науку (e-Social Science))

Современная **электронная наука (ЭН)** - наука, которая

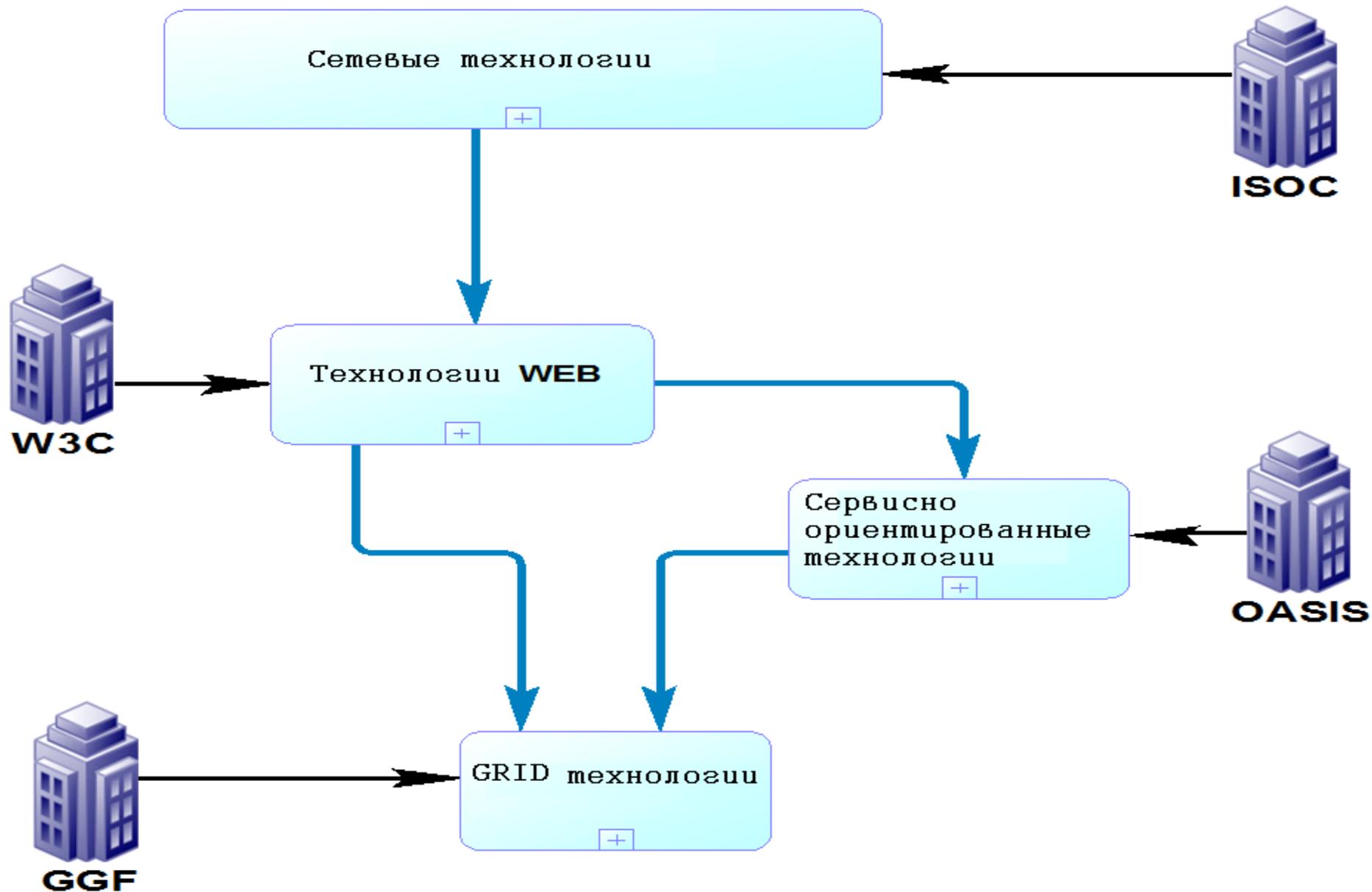
- а) развивается посредством глобального сотрудничества,
- б) связана с интенсивными вычислениями, которые выполняются в широко разветвленных распределенных высокоскоростных сетевых средах и используют мощные компьютерные ресурсы,
- в) использует сверхбольшие распределенные коллекции данных, подлежащие хранению, обработке, визуализации и обеспечению доступа к ним в реальном масштабе времени,
- г) основана на технологиях Интернет, Grid-вычислений (Grid computing) и глобального сетевого доступа.

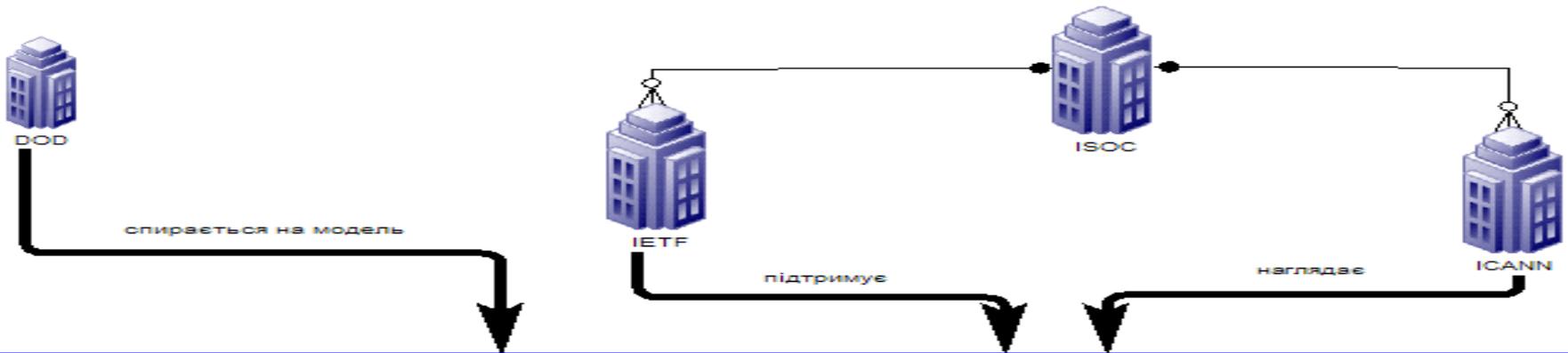
e-Наука в мире:

- **ЦЕРН** (CERN – European Organization for Nuclear Research) - Европейская организация по ядерным исследованиям. Проект ***EGEE II*** (Enabling Grids for E-science, "Развёртывание Grid для развития e-науки"). Более 70 учреждений из 27 стран мира, более 20 тыс. процессоров;
- **UK e-Science Programme** (Великобритания) – гос. программа;
- **Глобальный форум Grid** (GGF, Global Grid Forum) - разрабатывает *стандарты* для Grid-сообщества, ведет проект *Globus*, разрабатывает базовый инструментарий для построения Grid - *Globus Toolkit*;
- Другие проекты: ***Tera-Grid*** (США), ***DataGrid*** (Европа), ***China-Grid*** (Китай), ***GARUDA*** (Индия), ***GLORIAD*** - сеть в Северном полушарии (США, Канада, Европа, Россия, Китай, Южная Корея и др.).

Конференции и специализированные семинары:

- International e-Science (e-Science 2007),
- International Workshop on Scientific Workflows and Business Workflow Standards in e-Science (2006, 2007),
- IEEE Conf e- Science and Grid computing (2006, 2007),
- International Workshop on Workflow Systems in e-Science (WSES 2006, 2007).





TCP/IP

Прикладний рівень



Транспортний рівень



мережевий рівень



Канальний рівень



фізичний рівень

Сервисные ресурсы Интернет

- SOAP** протокол для определения форматов запросов к сетевым службам;
- WSDL** - язык сервиса MS.Net для обмена данными среды .NET и развития сетевых служб .Net Remoting;
- UDDI** (Universal Description, Discovery and Integration) для регистрации, описания, хранения, поиска в реестре;
- **BPMN** графический язык для нотации прикладных и бизнес-процессов типа UML; **BPEL** язык описания бизнес процессов;
- SOA** (Service-oriented Architecture) модель сервисно-ориентированной архитектуры программных систем;
- **SCA** (Service-Component Architecture) модель создания сложных систем на основе сервисов и компонентов и др.

Термин семантик-Web ввел Тим Бернес-Ли (2001) в журнале «**Scientific American**». Семантические основы Web (Веб) развиваются многими исследовательскими Институтами США и Евросоюза, а также множеством больших и малых производителей разных систем, проектов с открытым кодом. Проводятся более 10 лет конференции по Веб-семантическим ресурсам и технологиям (<http://knowledgeweb.semanticweb.org>, <http://semwebprogramming.com> и др.

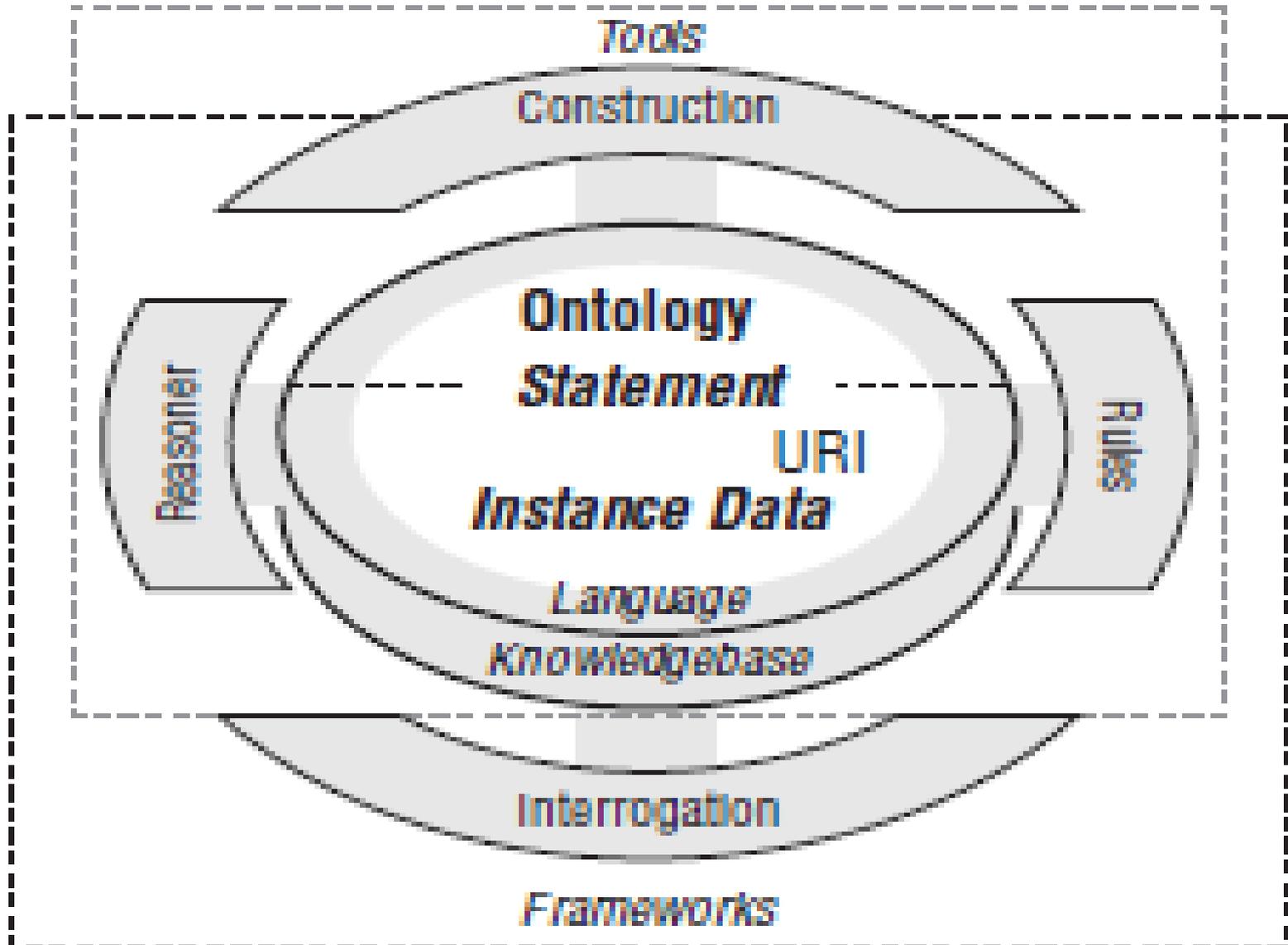
Слово «семантический» означает «**осмысленный**», «**понятный**». В сети Веб каждый ресурс задается человеческим языком, дополненным описанием в языках, понятных компьютеру. В мировой информационной сети (World Wide Web) Интернет содержится множество **семантических ресурсов**, средств и инструментов для разработки онтологий, систем и бизнес-процессов.

Semantic Web – является единым информационным пространством, где главными единицами информации выступают сущности и онтологии. Они описываются стандартами (отличными от привычных HTML и CSS), к которым относятся стандарты, разработанные всемирным консорциумом W3C - RDF, OWL, SparQL и др.

Описание онтологии на языке OWL – это последовательность аксиом и фактов, информации о классах, свойствах, ресурсах и документах веб, которые импортируются через URI и могут ссылаться на XML–схему по форме:
<Datatype ID>:: = < URI ссылки >.

К системами поддержки описаний онтологий относятся FODA, Protégé, DSL Tool и др.).

Базовые компоненты семантического веба



Основные понятия онтологий

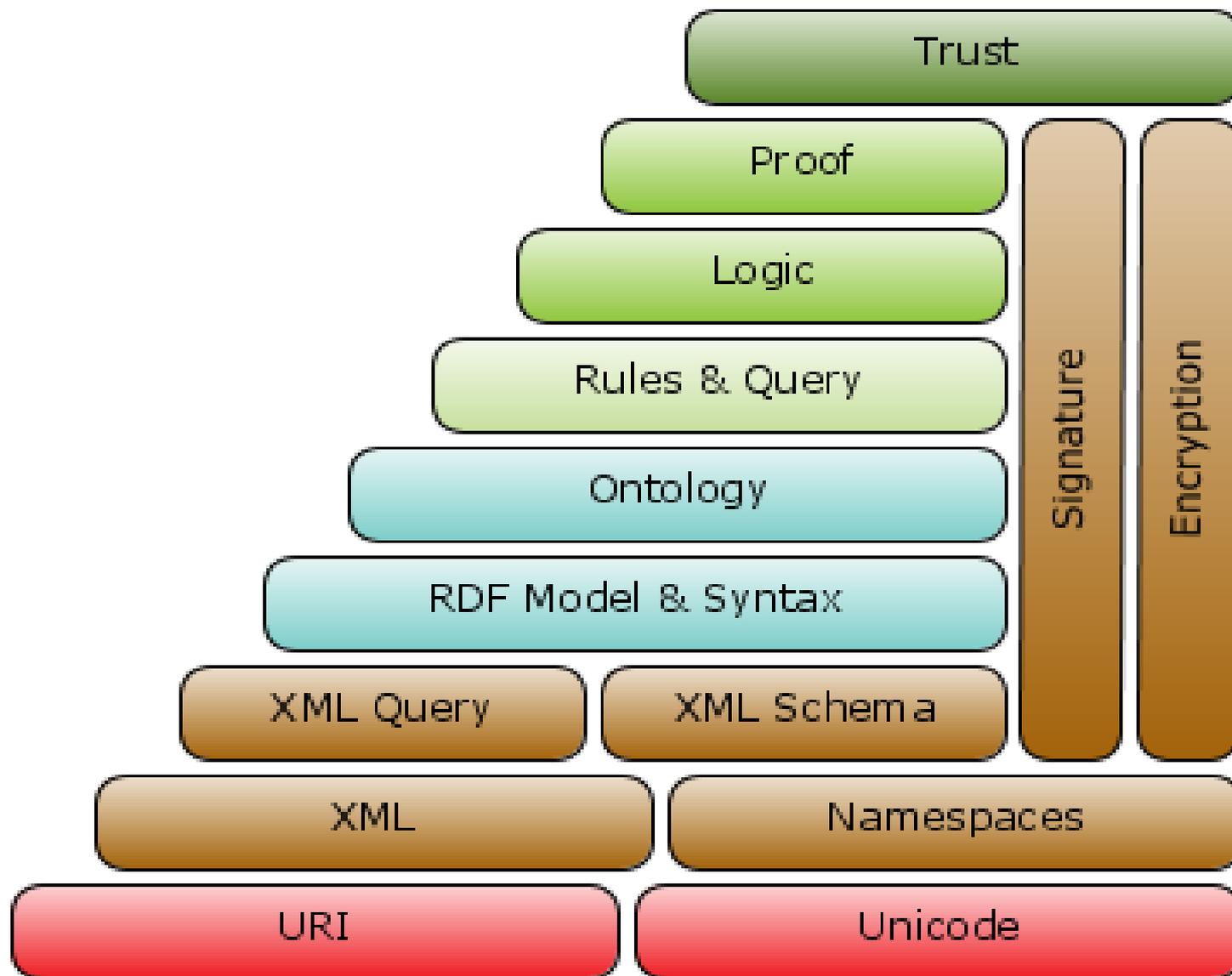
Это объекты в виде абстрактных групп, коллекций или наборов объектов, которые могут включать в себя индивиды, классы и их сочетания.

Объекты могут иметь атрибуты – имя и значение, которые используются для сохранения информации, специфической для каждого объекта.

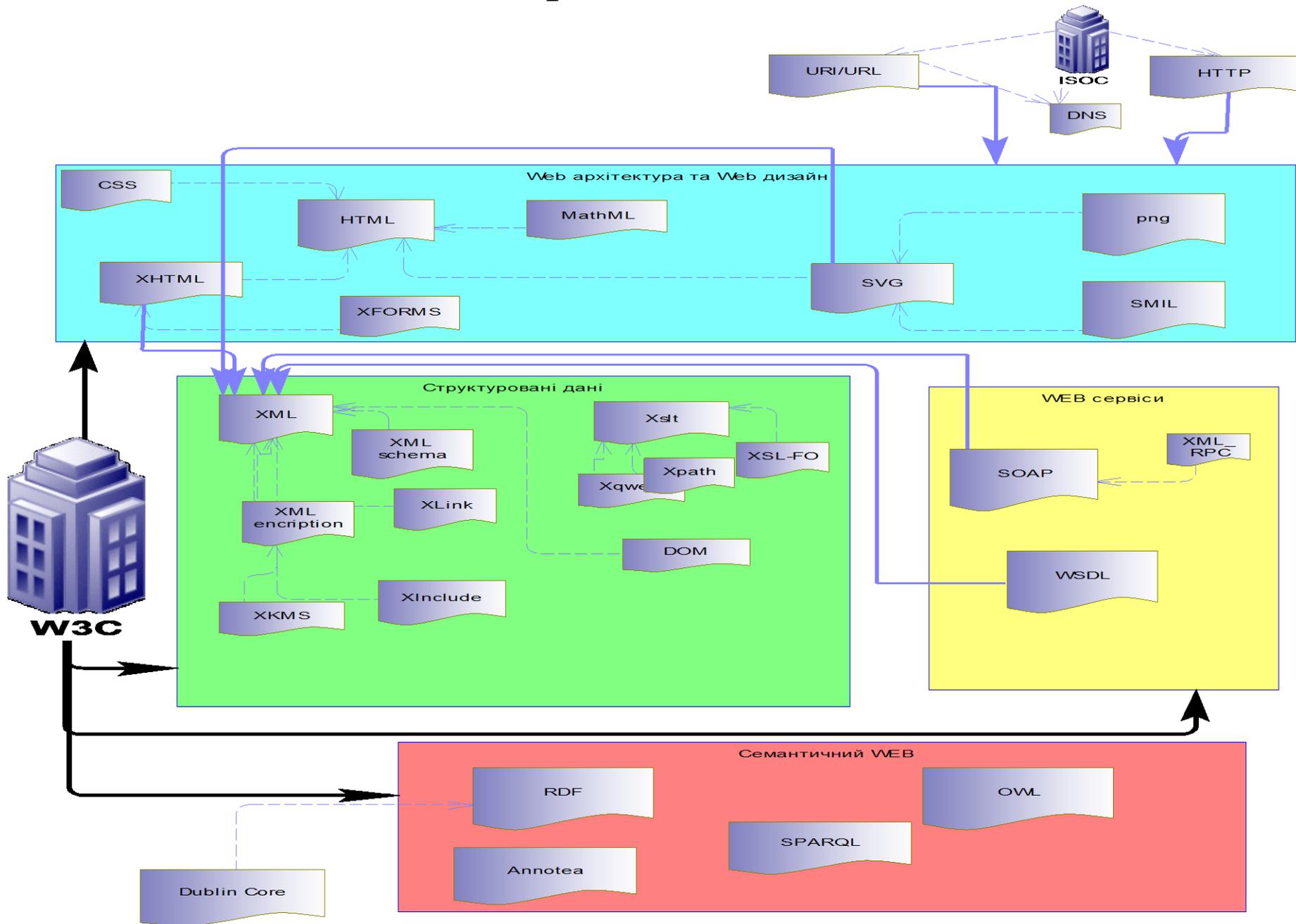
Модель предметной области описывается с помощью языков OWL, RDF и RDFs и KIF (Knowledge Interchange Format).

Виды технологий: редакторы онтологий (для создания онтологий), DBMS онтологий (для хранения и обращения к онтологии) и хранилища онтологий (для работы с разными онтологиями) и системы их поддержки (FODA, Protégé, DSL Tool и др.).

Средства для описания онтологических систем



Стандартные языки WEB



Характеристика Веб-языков

- **SOAP** протокол для определения форматов запросов к сети;
- **WSDL** - язык сервиса MS.Net для обмена данными среды .NET и развития сетевых служб .Net Remoting;
- **UDDI** (Universal Description, Discovery and Integration) для регистрации, описания, хранения, поиска в реестре;
- **BPMN** графический язык для нотации прикладных и бизнес-процессов типа UML; **BPEL** язык описания бизнес процессов;
- **SOA (Service-oriented Architecture)** модель сервисно-ориентированной архитектуры программных систем;
- **SCA (Service-Component Architecture)** модель создания сложных систем на основе сервисов и компонентов;
- **XML** язык разметки, описания структуры и способов взаимодействия компонентов;
- **HTML** язык разметки документов и гипертекстов для любых Web-браузеров и редакторов.
- **MathML** (Mathematical Markup Language) — язык математических формул на основе **XML** и др.

Элементы онтологий для задания доменов

Это классы, факты, аксиомы, фасеты, слоты и др.

Класс описывает понятия предметной области (ПрО). Он может быть эквивалентом, подмножеством или пересечением более общих классов с ограничениями. Классы могут иметь собственные слоты. Например, документация класса - собственный слот, присоединенная к классу;

Слот – задает свойства (атрибуты) и ограничения для типов значений (целое, символьное и др.) с границами (min, max). Слот может быть *template slot*, *own*, присоединяется к фрейму.

Фасеты задают свойства слотов (конкретные типы и возможные диапазоны значений) и ограничение на ТД для присоединения слота к фрейму класса подобно XML-схемы.

Аксиома класса определяет совокупность описаний, которые могут иметь вид обобщенных классов, ограничений, наборов ресурсов, булевых комбинаций описаний и др.

- **XML** язык разметки, описания структуры и способов взаимодействия компонентов;
- **HTML** язык разметки документов и гипертекстов для любых Web браузеров и редакторлв.
- **MathML** (Mathematical Markup Language) — язык математических формул на основе **XML**;
- **SVG** (Scalable Vector Graphics) — язык масштабируемой векторной графики в формате **XML**;
- **EXHTML** (Extensible Hypertext Markup Language) – язык, который воспроизводит возможности языка HTML;
- **IContact** WCF сервисы MS.Net для обмена данными в среде .NET, как развитием сетевых служб, .Net Remoting.

Сервисно-ориентированная архитектура (SOA)

Модель SOA – это набор принципов и средств создания системного ПО и прикладных программных систем (ПС) из совместимых и унифицированных сервисов.

К принципам **SOA** относятся:

- независимость сервисов от вычислительной платформы и ЯП;
- единообразие интерфейса при построении Веб-систем.

К средствам работы с сервисом относятся:

- локатор сервиса (**service locator**) для поиска веб-сервиса;
- локатор интерфейса;
- stub для связи (**SOAP binding stub**) парсинга SOAP;
- интерфейс сервиса;
- прокси-класс, который реализует интерфейс;
- клиентский стаб и локатор для доступа к веб-сервисам.

Сервисно-компонентная архитектура (SCA)

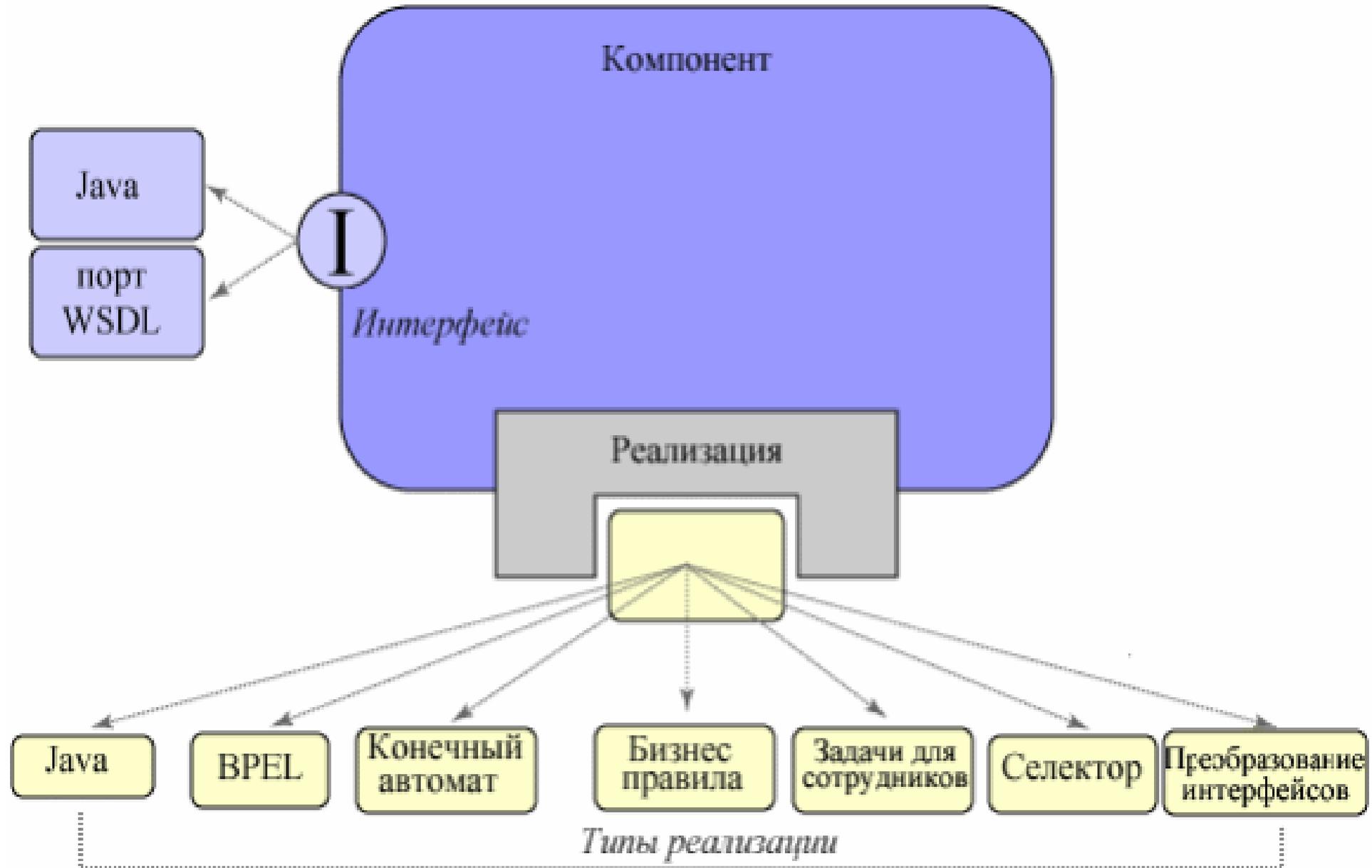
SCA обеспечивает доступ к готовым сервисным компонентам системы IBM **WebSphere** Integration Developer (на платформе Eclipse) могут быть упакованы в модуль для выполнения сервиса в WebSphere.

Process Server – эквивалентен EAR-файлу J2EE. Подмодули **J2EE** упаковываются с **SCA**, запускают сервис и передают данные. **SCA** модуль обращается к **Enterprise JavaBean**.

Обмен данными проводится через **SDO** (**Service Data Objects**),

В **SCA** сервисы могут собираться в различные образования (**хореографии**). Они в **SOA** и **SCA** создают новые сервисы, задают их комбинации и конфигурации.

Общая схема сервера SCA IBM



Веб-сервисы и ASP.NET

Веб-системы создаются и в архитектуре **ASP.NET**:

- для вызова веб-сервиса используется серверная технология **AJAX**;

- обмен данными между клиентом и сервером осуществляется в формате **JSON (POST HTTP и GET HTTP)**.

В платформе **ASP.NET** :

- создается файл с расширением **ASMX** и с атрибутом **WebService** (точка входа);

- в описании класса задаются открытые методы с атрибутом **WebMethod**;

- функциональность Веб-сервиса это **ASMX-файл** с передкомпилированными сборками.

Технология Jasp

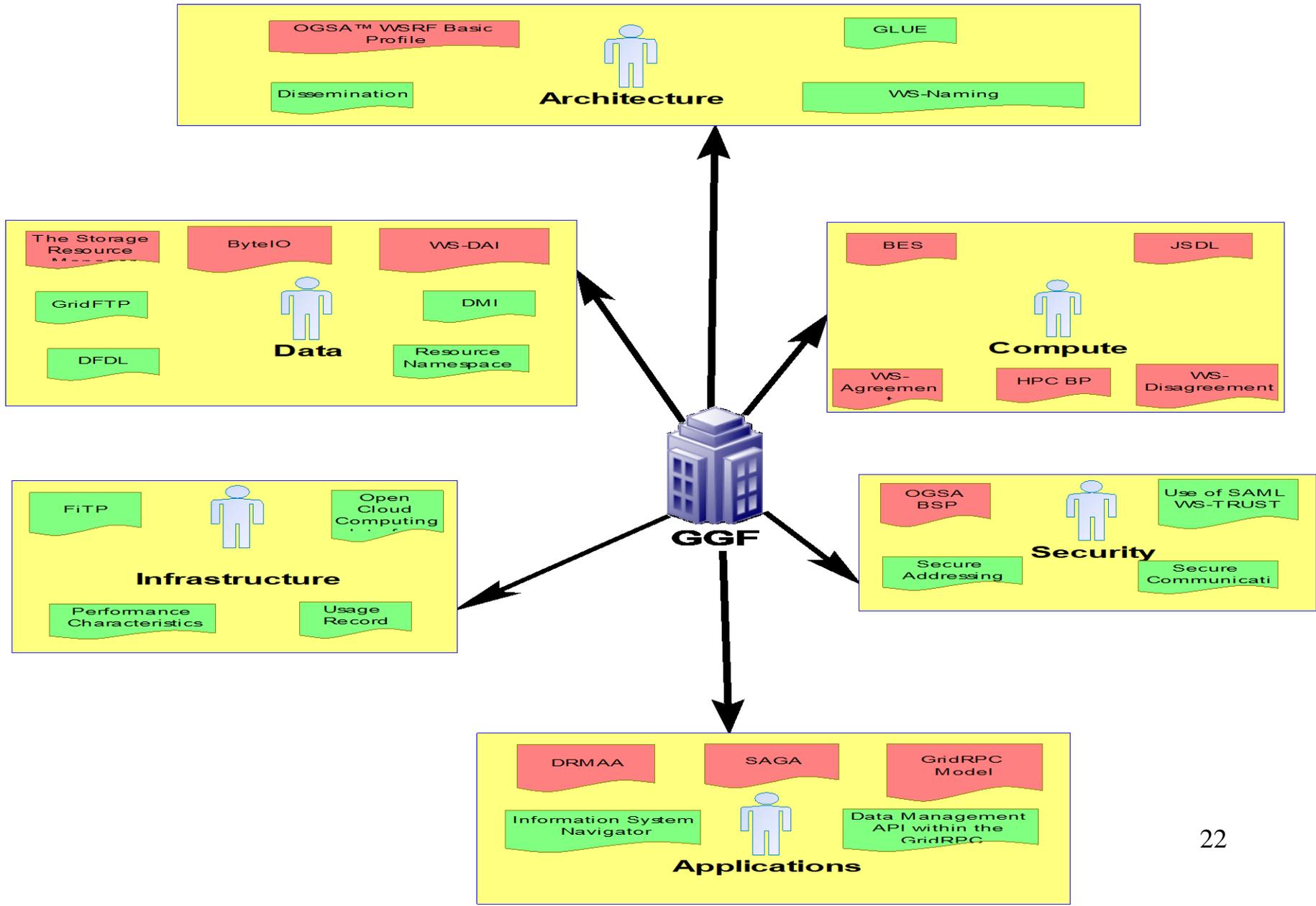
Jasp - это средства для разработки Веб-систем в Microsoft.Net.

В этой платформе обеспечивается:

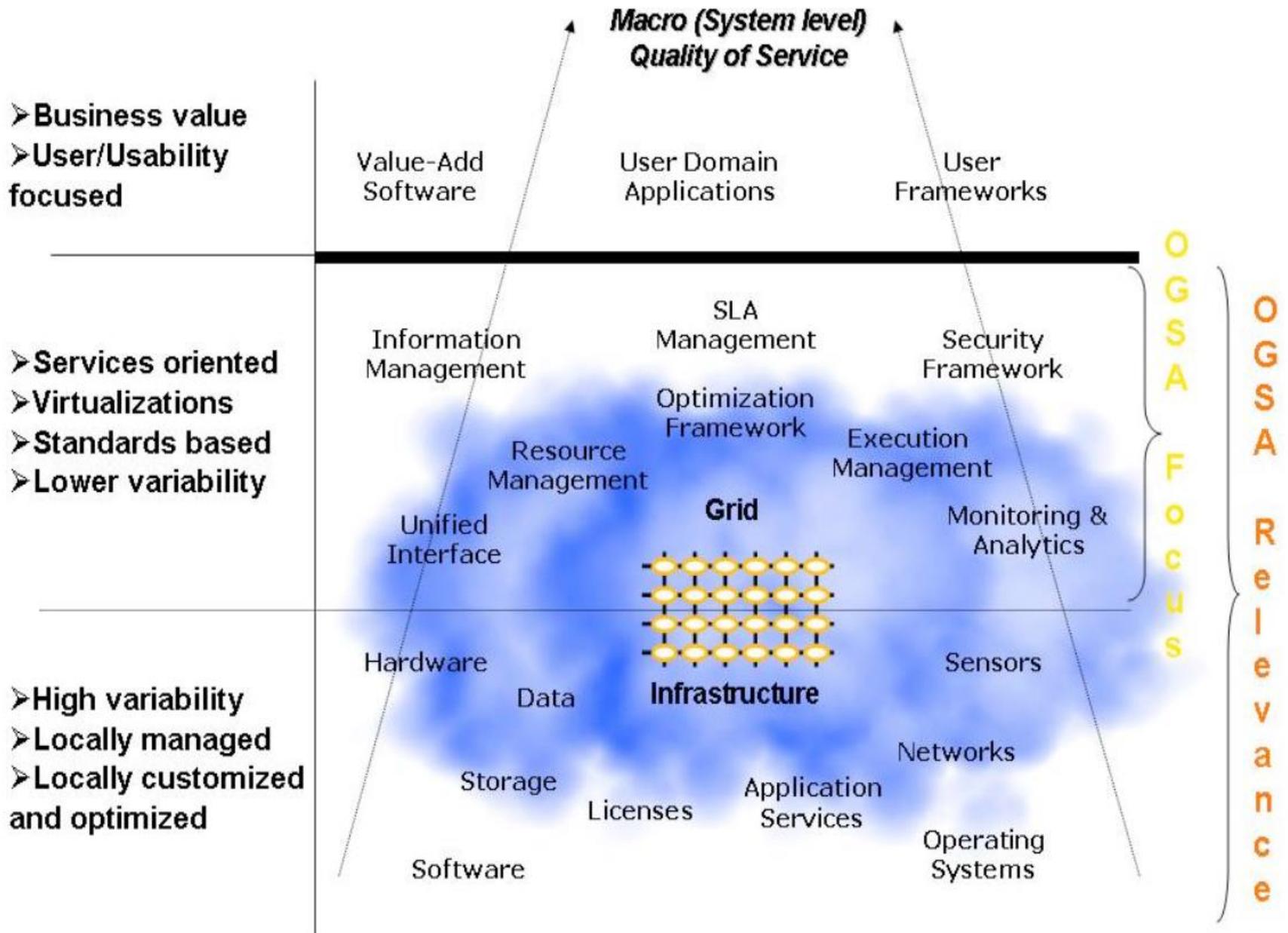
- повторное использование компонентов, объектов;
- наследование элементов в серверной и клиентской частях;
- **JavaScript, CSS** и разметки;
- автоматическая загрузка ресурсов (синхронно или асинхронно);
- работа с JavaScript библиотеками: jQuery, ExtJS, Prototype и т.п.

Jasp определяет следующие ресурсы: **CSS, JavaScript** – **HTML** и бинарные ресурсы. Группирует все CSS и JavaScript файлы в один CSS или JavaScript файл и добавляет специальные ссылки HTML разметки для загрузки браузером.

Grid технологии



Общая схема Grid



Структура сервисов промежуточного слоя Grid

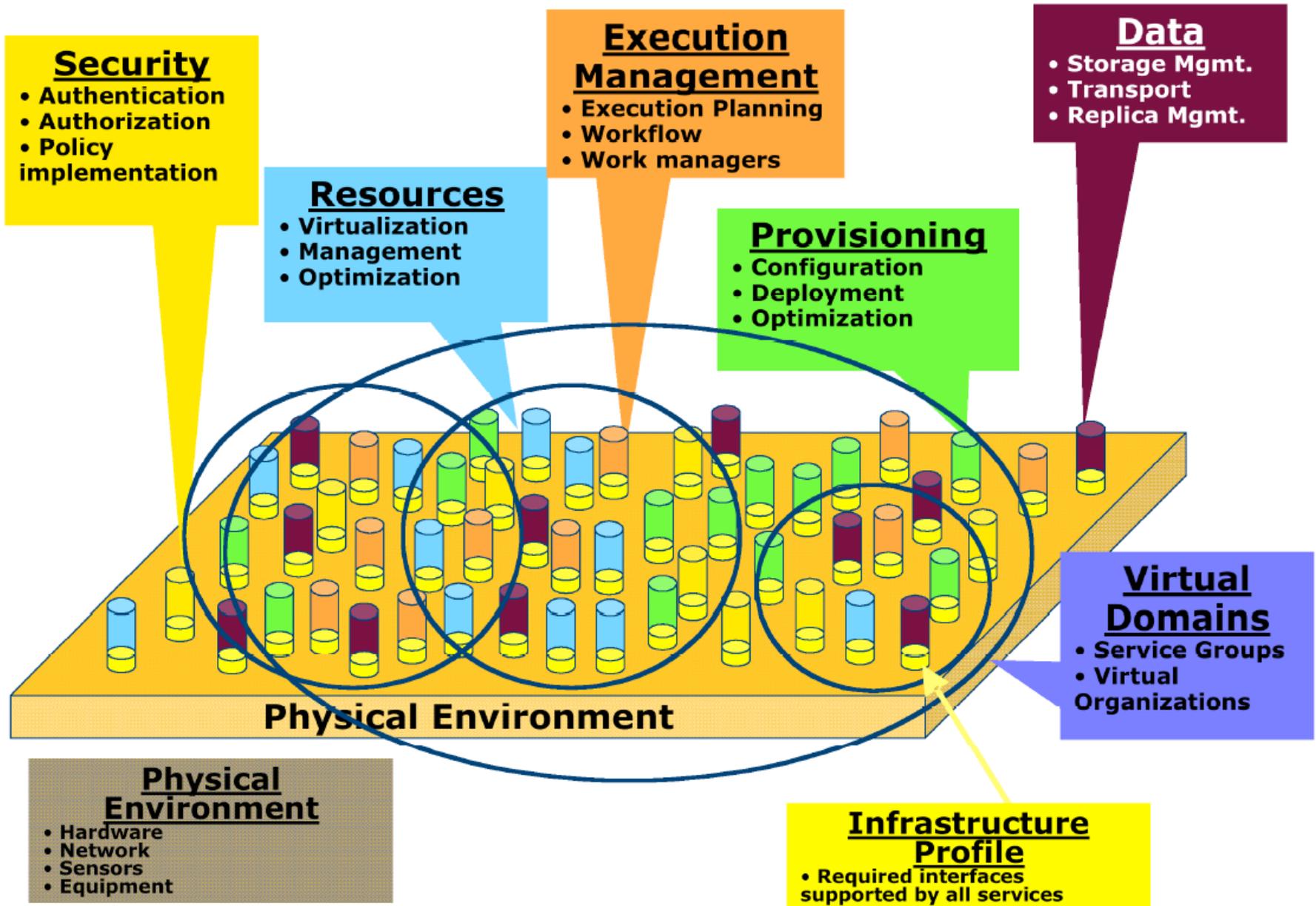
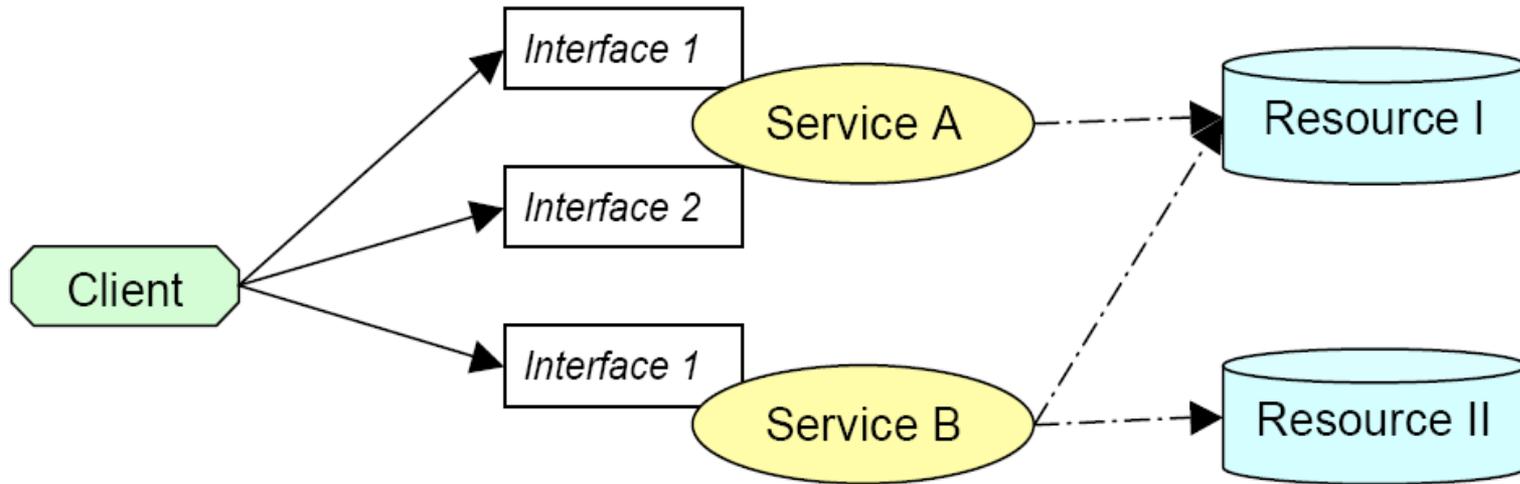


Схема доступа к ресурсам Grid



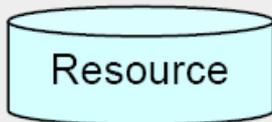
Key:



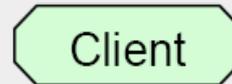
An interface of a service



A service



A data resource



A client application or service

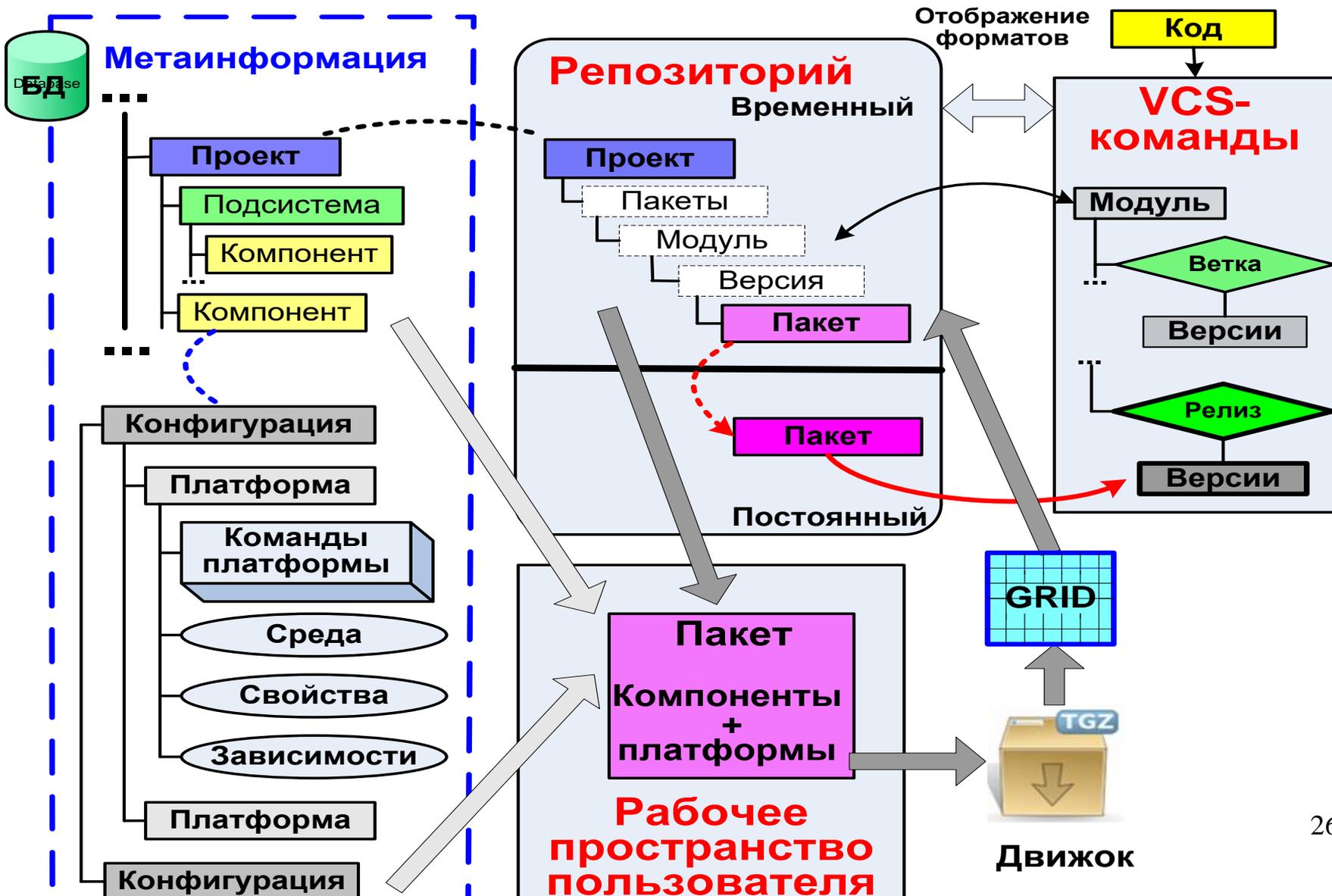


An API or service calling an interface



A service using a resource

Технология создания прикладных систем в Grid



Различие между Grid-сервисами и Web-сервисами. Виртуализация

Web-сервисы - не имеют состояния. Срок жизни определяется поставщиком услуги. Существуют независимо от потребителя.

Grid-сервисы

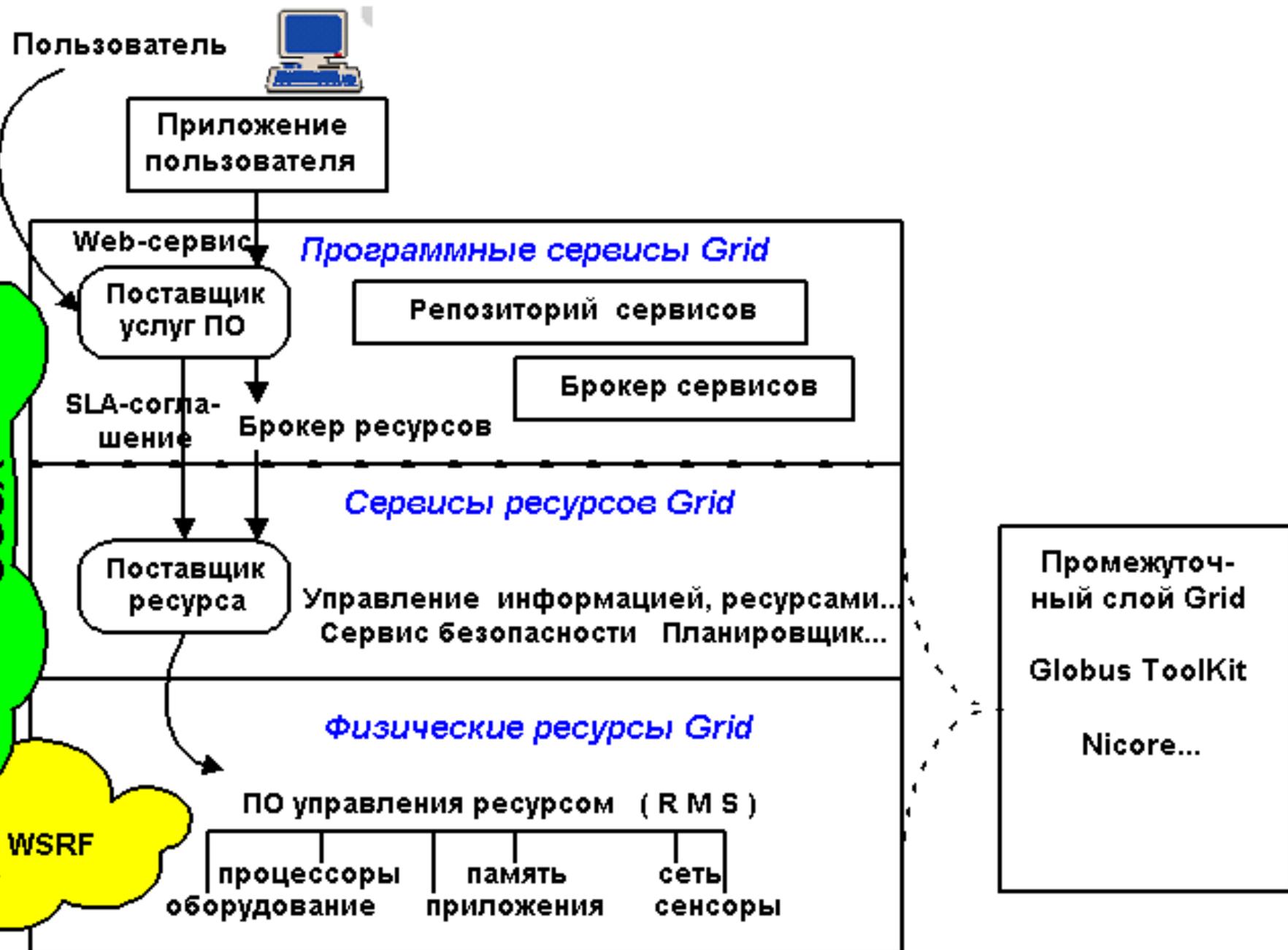
- имеют **состояние** и **срок жизни**, ограниченный *потребностями «потребителя»*,
- «рождаются» на серверах в Grid-сети для решения **конкретной задачи** и «исчезают» по ее завершении.
- представляют «свой» сервер как **фабрику**, на которой **по заказу** клиента создается нужный сервис, используемый в заданное время.

Этот подход называется «**фабрика образцов**» (factory).

Сам сервер (=сервер-посредник) услуг не оказывает. **Фабрикует** нужный «образец» услуг, который оформляется как **сетевой сервис (виртуальный)**.

Если возникает необходимость решить какую-то задачу в Grid-сети, под нее создается **виртуальный компьютер**, предоставляющий нужный **физический сервер**. Этот сервер может поддерживать множество подобных **виртуальных вычислительных машин**, создаваемых под требования клиента и уничтожаемых, когда в них отпадает необходимость.

Обобщенная архитектура Grid



Архитектура протоколов Grid



Базовый уровень (Fabric Layer) описывает сервисы, непосредственно работающие с сетевыми ресурсами (выч-ми, инф-ми, памяти).

Уровень связи (Connectivity Layer) определяет коммуникационные протоколы и протоколы аутентификации.

Ресурсный уровень (Resource Layer) реализует протоколы работы с отдельно взятыми ресурсами.

Коллективный уровень (Collective Layer) отвечает за глобальную интеграцию различных наборов ресурсов (обнаружение, выделение и др.).

Прикладной уровень (Application Layer) описывает пользовательские приложения, работающие в среде виртуальной организации.

Научный процесс как поток научно-исследовательской работы

Среды формирования и исполнения потоков работ (workflows, WF) - разновидность проблемно-ориентированных сред.

Поток научно-исследовательской работы = **научный процесс**

Схема научного процесса (*Scientific workflow*) - описание набора научно-исследовательских работ и порядка их выполнения, обусловленного логикой обработки получаемой информации.

Отличия научного процесса (НП) от делового процесса (ДП):

Научный процесс - *управляем данными* (информацией, знаниями), передаваемыми от одного компонента процесса к другому.

Деловой процесс - *управляем целями* и *событиями*.

НП более гибкий, чем ДП, и может уточняться по ходу решения задач.

НП может быть остановлен в любой момент, в то время как ДП должен достичь приемлемого конечного состояния.

НП в большей степени, чем ДП, связан с неопределенностью и исключениями, поскольку управляем промежуточными результатами исследований.

Мотивация реализации НП – контролирование экспериментов и доведение до научной общественности информации о том, как они проводились (постановках, условиях, ограничениях, результатах).

Электронная наука: рецепт слоеного пирога



Нижний слой - наработки в сфере информатизации науки.

Промежуточные слои – наработки в сфере Web- и Grid-сетей (включая сети знаний).

Верхний слой (окутывающий) – средства целостной автоматизированной поддержки процессов научной деятельности.

Развитие ЭН - в рамках каждого слоя (не выходя за его границы) или по разным аспектам ЭН (вертикальными срезами).

2. Средства описания онтологий в Семантик Веб

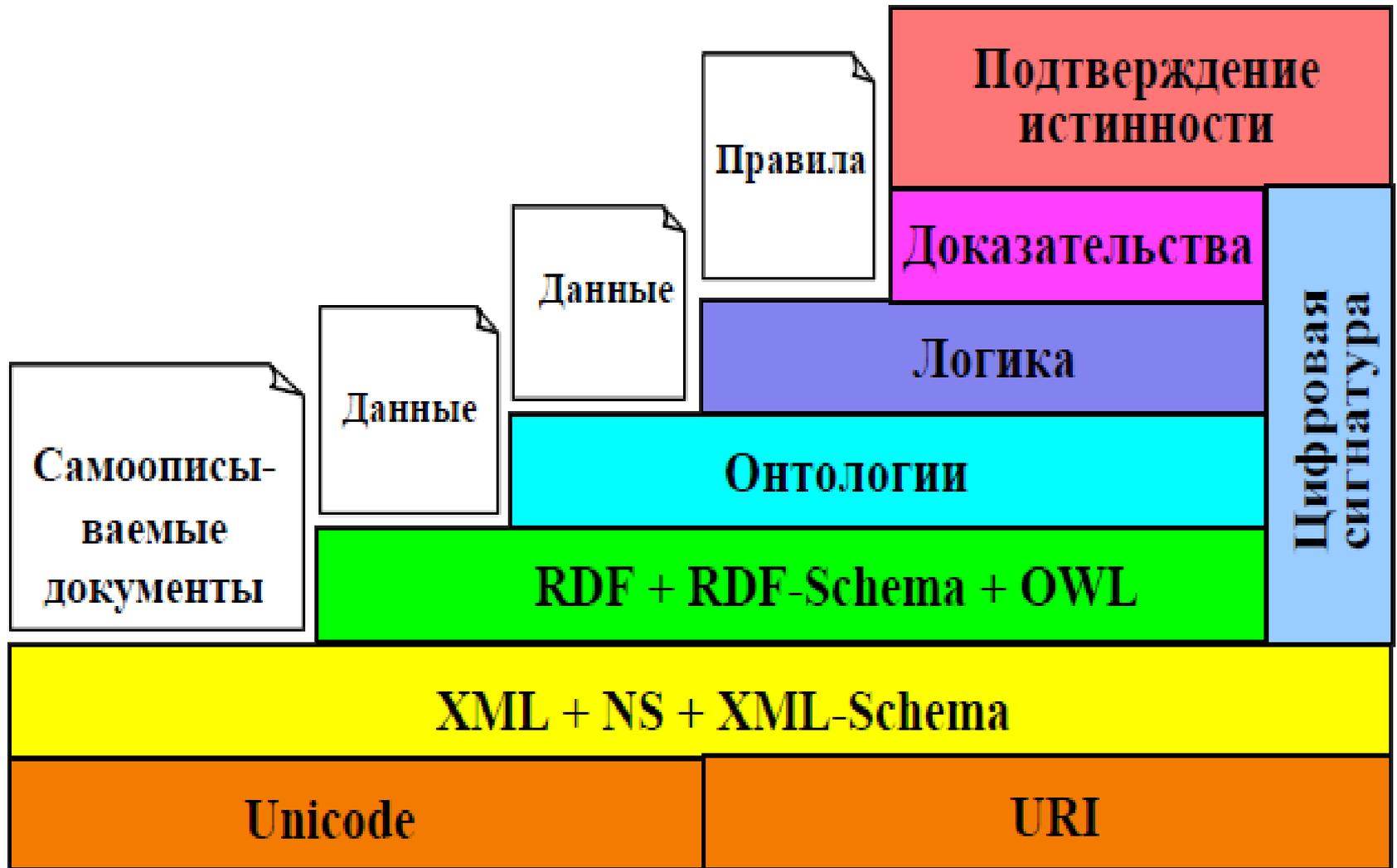
К ним относятся объекты, абстрактные группы, коллекции или наборы объектов, которые могут включать в себя индивиды, классы и их сочетания.

Объекты могут иметь атрибуты – имя и значение, которые используются для сохранения информации, специфической для каждого объекта.

Концептуальная **Модель** предметной области задается языками OWL, RDF и RDFs и KIF (Knowledge Interchange Format).

Виды технологий: редакторы онтологий для создания онтологий, DBMS онтологий для хранения и обращения к онтологии и хранилища онтологий для работы с разными онтологиями) и системы их поддержки (FODA, Protégé, DSL Tool и др.).

Схема средств описания средств Семантик Веб



Языки описания в Семантик Веб

- **SOAP** протокол для определения форматов запросов к сети;
- **WSDL** - язык сервиса MS.Net для обмена данными среды .NET и развития сетевых служб .Net Remoting;
- **UDDI** (Universal Description, Discovery and Integration) для регистрации, описания, хранения, поиска в реестре;
- **BPMN** графический язык для нотации прикладных и бизнес-процессов типа UML; **BPEL** язык описания бизнес процессов;
- **SOA (Service-oriented Architecture)** модель сервисно-ориентированной архитектуры программных систем;
- **SCA (Service-Component Architecture)** модель создания сложных систем на основе сервисов и компонентов;
- **XML** язык разметки, описания структуры и способов взаимодействия компонентов;
- **HTML** язык разметки документов и гипертекстов для любых Web-браузеров и редакторов.
- **MathML** (Mathematical Markup Language) — язык математических формул на основе **XML** и др.

Действующие онтологии областей знаний

В мировой практике работает ряд действующих онтологий:

- Sensus базируется на понятиях естественного (английского) языка и содержит более чем 70 000 терминов и их дефиниций;

- понятийный сборник электронной коммерции;

- глобальные продукты и услуги (Организации Объединенных наций);

- SCTG коммерческие транспортные потоки товаров компаний;

- e-cl@ss поддерживает обмен данными и материалами между продавцами и пользователями крупных компаний Германии;

- товары RosettaNet поддерживают 400 коммерческих компаний;

- требования в программной инженерии;

- жизненный цикл (ЖЦ) компьютерных систем;

- вычислительная геометрия <http://cg.inicyb.kiev.ua>

- computational geometry.

Базовые элементы онтологий в Семантик Веб - классы, факты, аксиомы, фасеты, слоты и др.

Класс описывает понятия предметной области (ПрО).

Понятие может быть подмножеством или пересечением более общих классов с ограничениями. Классы могут иметь собственные слоты. Например, документация класса - собственный слот, присоединяемый к классу;

Слот – задает свойства (атрибуты) и ограничения для типов значений (целое, символьное и др.) с границами (min, max).

Слот может быть *template slot*, *own*, который присоединяется к фрейму.

Фасеты задают свойства слотов (конкретные типы и возможные диапазоны значений) и ограничение на типы данных для присоединения слота к фрейму класса как XML-схема.

Аксиома класса определяет совокупность описаний, которые могут иметь вид обобщенных классов, ограничений, наборов ресурсов, булевых комбинаций описаний и др.

Онтологии - вычислительной математики и в программной инженерии

1. Вычислительная геометрия

**2. Жизненный цикл программных и
информационных систем задается
стандартом ISO/IEC 12207 Life Cycle (ЖЦ)**

1. Онтология вычислительной геометрии

Термин "вычислительная геометрия» определили Препарати и Шеймос (1975) и включает разделы:

- комбинаторная вычислительная геометрия или алгоритмическая геометрия, которая рассматривает геометрические объекты как дискретные сущности.
- численная вычислительная геометрия или машинная геометрия, геометрическое моделирование представления объектов реального мира в форме пригодной для дальнейшей компьютерной обработки или раздел компьютерной графики.

Вычислительная геометрия. Статические задачи

- *Выпуклая оболочка*: Имея набор точек, необходимо найти наименьший выпуклый многоугольник, который содержит все точки;
- *Пересечение отрезков*: найти все пересечения в наборе отрезков;
- *Триангуляция Делона*;
- *Диаграмма Вороного*: Имея набор точек, разделить пространство на сектора, каждая точка которых ближе к ней из набора других;
- *Задача ближайшей пары точек*: Имея набор точек найти кратчайшее расстояние;
- *Евклидов кратчайший путь*: Соединить две точки Евклидова пространства (с полигональными препятствиями) кратчайшим образом.
- *Триангуляция многоугольника*: Имея многоугольник, разбить его на внутренние треугольники.

Вычислительная геометрия. Задачи поиска

– *Региональный поиск.* Обработать набор точек, с целью эффективного поиска набора точек содержащейся в запрошенном регионе.

– *Локализация точки.* Имея разбиение пространства на регионы, создать структуру данных, что позволит эффективно определить в каком регионе находится данная точка.

– *Поиск ближайшего соседа.* Обработать набор точек чтобы иметь возможность эффективно найти точки к запрошенной.

– *Трассировка лучей.* Для заданного набора объектов в пространстве создается структура данных, которая позволяет эффективно определять объекты, которые пересекают запрошенный луч.

Вычислительная геометрия. Описание в Protégé 4.1

Платформа Protege поддерживает два способа моделирования онтологий с помощью редактора Protege-Frames и Protege-OWL.

Онтологии, построенные в Protege, могут быть экспортированы во множество форматов, включая RDF (RDF Schema), OWL и XML Schema.

Онтология домена «Вычислительная геометрия» включает набор понятий и связей между ними, которые зафиксированы в схеме онтологии (ниже) и отображены на сайте <http://cg.incub.kiev.ua>

Схема онтологии домена «Вычислительная геометрия» после Protege

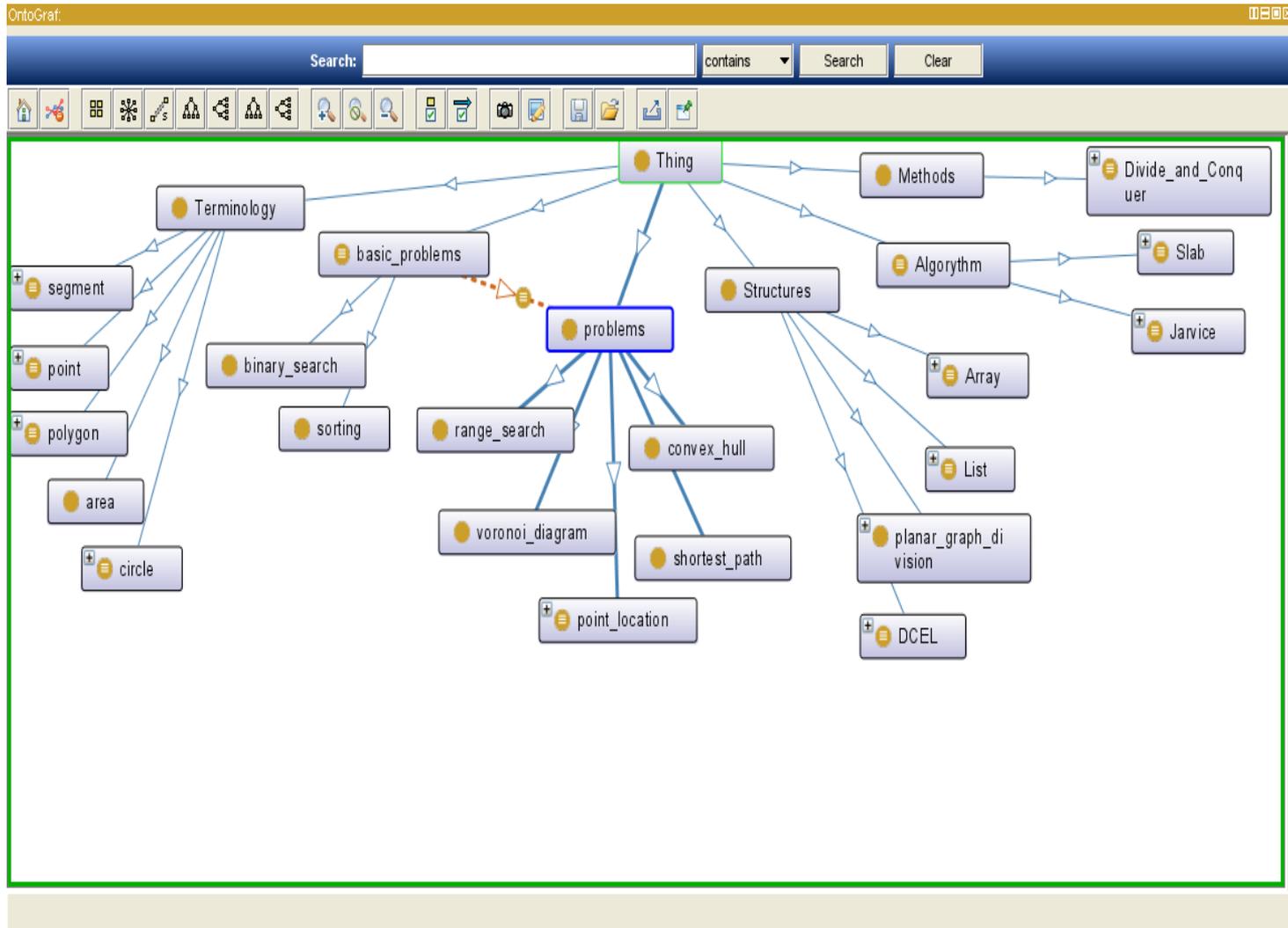


Схема выполнения онтологии Вычислительная геометрия на сайте:
<http://7dragons.ru/ru>

Название

Ontology1324927182924.owl

Размер 16.10 kB

Дата модификации

12.07.2015, 20:00:03

Описание

**Описание онтологии предметной области
«Вычислительная геометрия» в формате OWL.**

2. Онтология домена – стандарт ЖЦ 12207

Стандарт включает:

- 1) основные процессы;
- 2) процессы поддержки;
- 3) организационные процессы.

Все процессы включают процессы (17), действия (activity) 74, задачи (task) 232.

Процессы, подпроцессы и задачи ЖЦ 12207

Классы	Процесс	Действие	Задача
Основные процессы	5	35	135
Процессы поддержки	8	25	35
Организационные процессы	4	14	27
Всего	17	74	232

Основные процессы

1. Категория «Основные процессы»

1.1 Заказ (договор)

1.1.1 Подготовка заказа, выбор поставщика

1.1.2. Мониторинг деятельности поставщика, принятие потребителем

1.2. Поставка (приобретение)

1.3. Разработка

1.3.1. Выявление требований

1.3.2. Анализ требований к системе

1.3.3. Проектирование архитектуры системы

1.3.4. Анализ требований к системе

1.3.5. Проектирование системы

1.3.6. Конструирование (кодировка) системы

1.3.7. Интеграция системы

1.3.8. Тестирование системы

1.3.9. Системная интеграция

1.3.10. Системное тестирование

1.3.11. Инсталляция системы

1.4. Эксплуатация

1.4.1. Функциональное применение

1.4.2. Поддержка потребителя

1.5. Сопровождение

Фрагмент описания основных процессов в XML

```
<?xml version="1.0" encoding="utf-8"?>  
<AssociationLine Name = "Определение требований"  
Type="Main. Определение требований" ManuallyRouted  
= "true" FixedFromPoint = "true" </AssociationLine>  
<AssociationLine Name = "Интеграция_ПС" Type =  
"Main.Интеграція_ПС"  
<AssociationLine Name = "Инсталляция" Type =  
"Main.Інсталяція"  
<AssociationLine Name = "Эксплуатация" Type =  
"Main.Експлуатація"  
<Property Name = "Интеграция_ПС" />  
<Property Name = "Инсталляция" />  
<Property Name = "Анализ требований" />  
<Property Name = "Эксплуатация" />...
```

Процессы поддержки в ЖЦ

2. Категория «Процессы поддержки»

2.1. Документирование

2.2. Управление конфигурацией

2.3. Обеспечение гарантии качества

2.4. Верификация

2.5. Валидация

2.6. Общий обзор

2.7. Аудит

2.8. Решение проблем

2.9. Обеспечение применимости продукта

2.10. Оценивание продукта

Схема процессов поддержки

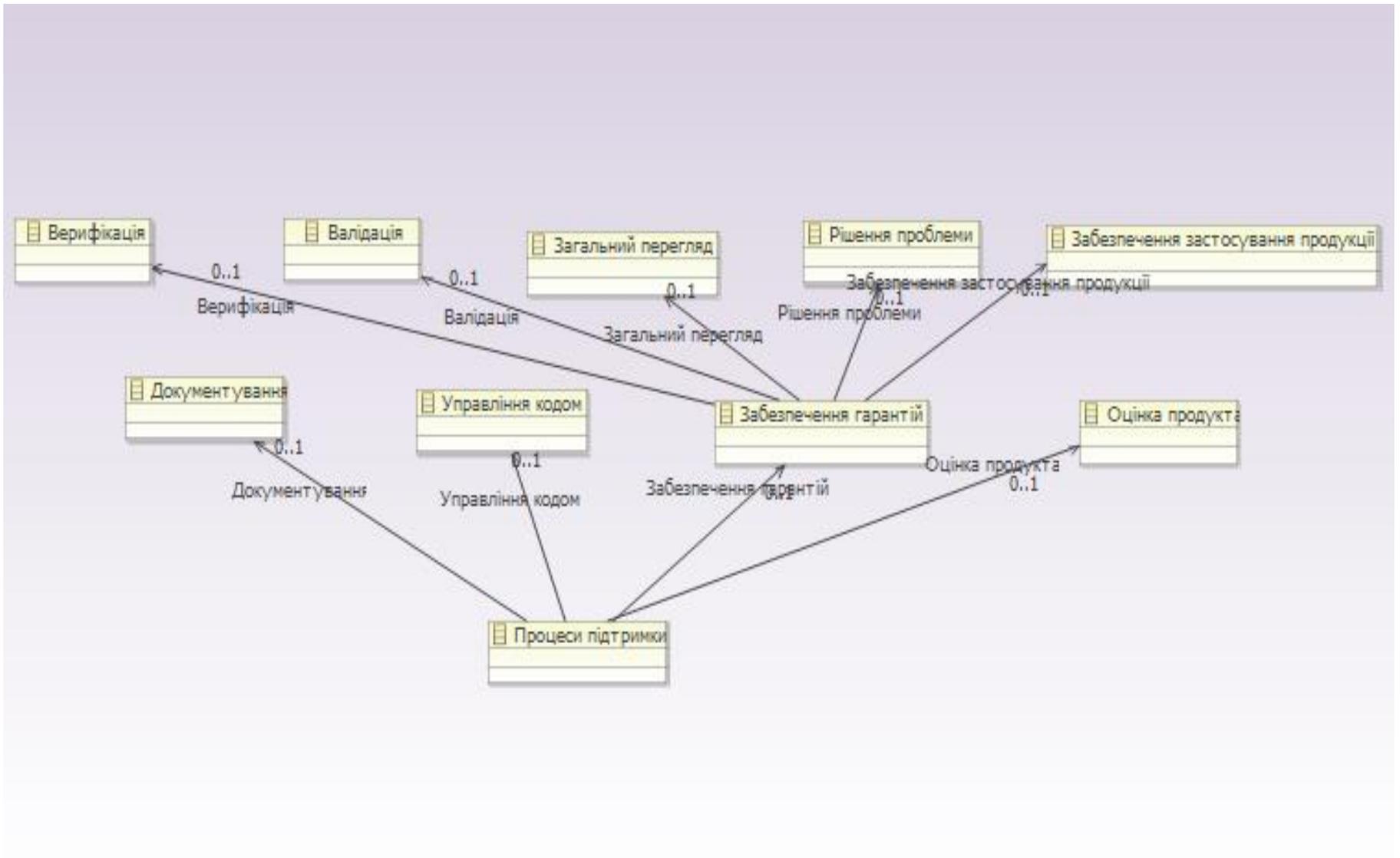
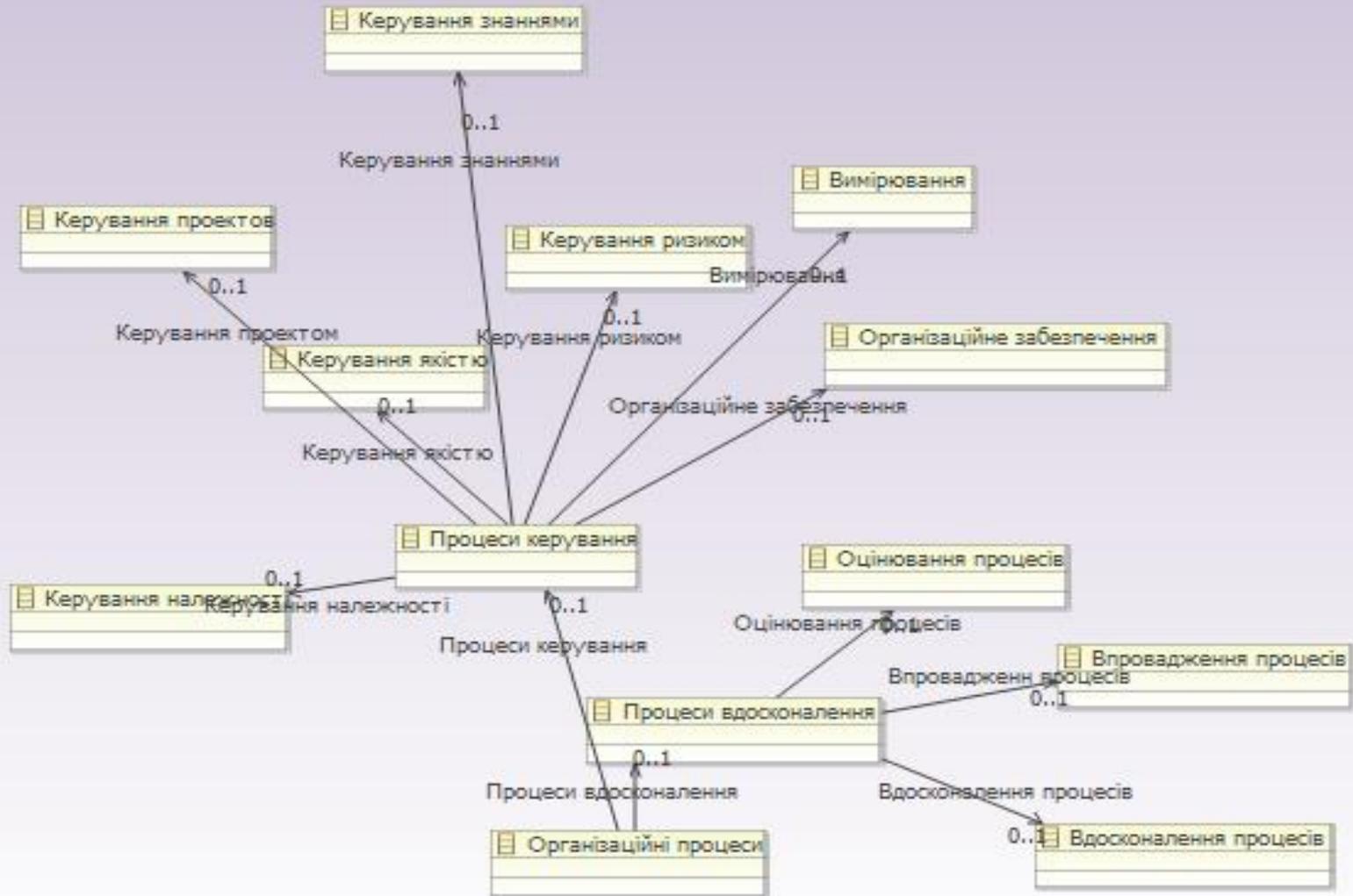


Схема процессов организационных процессов



Ontological approach to the formal specification of the Standard Life Cycle ISO/IEC 12207

Science and Information Conference-2015,
26-31 July, London



E.M. Lavrischeva,
doctor of phy. and math. sciences,
Honoured worker of science and technique
Ukraine, prof. MIPT, Main scientist of ISP
RAS

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2018615442

«Реализация метода онтологического моделирования домена
ЖЦ стандарта ISO/IEC 12207»

Правообладатель: *Федеральное государственное бюджетное
учреждение науки Институт системного программирования
им. В.П. Иванникова Российской академии наук (RU)*

Автор: *Лаврищева Екатерина Михайловна (RU)*

Заявка № 2018612585

Дата поступления 20 марта 2018 г.

Дата государственной регистрации
в Реестре программ для ЭВМ 08 мая 2018 г.

Руководитель Федеральной службы
по интеллектуальной собственности

 Г.П. Ильев



Процесс представления знаний в E-science при построении систем

Процесс представления *знаний* задается методологиями, основанными на знаниях (Knowledge-based systems, KBS), Common KADS с использованием библиотек, содержащих элементы решения задач ПрО с reuses, а также агентной методологией мультиагентного анализа и проектирования систем и др.

К базовым сервисам знаний относятся процессы:

- **извлечение знаний** согласно образцам в CBS, CBR;
- **кластеризация и индексация** больших объемов информации;
- **связи одних сущностей** онтологий с другими;
- **динамическое аннотирование** контента для связи с концептуальной схемой;
- **уточнение** больших объемов контента;
- **визуализация** больших наборов контента по требованиям пользователя;
- **умозаключения и мониторинг**, диагностика тех или иных событий, выявление ситуаций и оценивание успешного завершения процесса.

3. Создание Веб-систем из сервисов Интернет

Создание Веб-систем –это научный процесс в E-Science.

К средствам работы с сервисами относятся:

- локатор сервиса (**service locator**) для поиска веб-сервиса;
- локатор интерфейса;
- stub для связи (**SOAP binding stub**) для парсинга SOAP;
- интерфейс определения сервиса;
- прокси-класс, который реализует интерфейс;
- клиентский стаб и локатор для доступа к веб-сервисам;
- использование **SOA** и **SCA**.

Модель SOA, SCA – это наборы принципов и средств создания системного ПО и прикладных программных систем (ПС) из совместимых и унифицированных сервисов Интернет.

К принципам относятся:

- независимость сервисов от платформы вычисления в ЯП;
- единообразность интерфейсов при построении Веб-систем.⁵⁴

Научный процесс как поток научно-исследовательской работы

Среды формирования и исполнения потоков работ (workflows, WF) - разновидность проблемно-ориентированных сред.

Поток научно-исследовательской работы = **научный процесс**

Схема научного процесса (*Scientific workflow*) - описание набора научно-исследовательских работ и порядка их выполнения, обусловленного логикой обработки получаемой информации.

Отличия научного процесса (НП) от делового процесса (ДП):

Научный процесс - *управляем данными* (информацией, знаниями), передаваемыми от одного компонента процесса к другому.

Деловой процесс - *управляем целями* и *событиями*.

НП более гибкий, чем ДП, и может уточняться по ходу решения задач.

НП может быть остановлен в любой момент, в то время как ДП должен достичь приемлемого конечного состояния.

НП в большей степени, чем ДП, связан с неопределенностью и исключениями, поскольку управляем промежуточными результатами исследований.

Мотивация реализации НП – контролирование экспериментов и доведение до научной общественности информации о том, как они проводились (постановках, условиях, ограничениях, результатах).

Задачи научного процесса

Научный процесс – процесс получения нового знания.

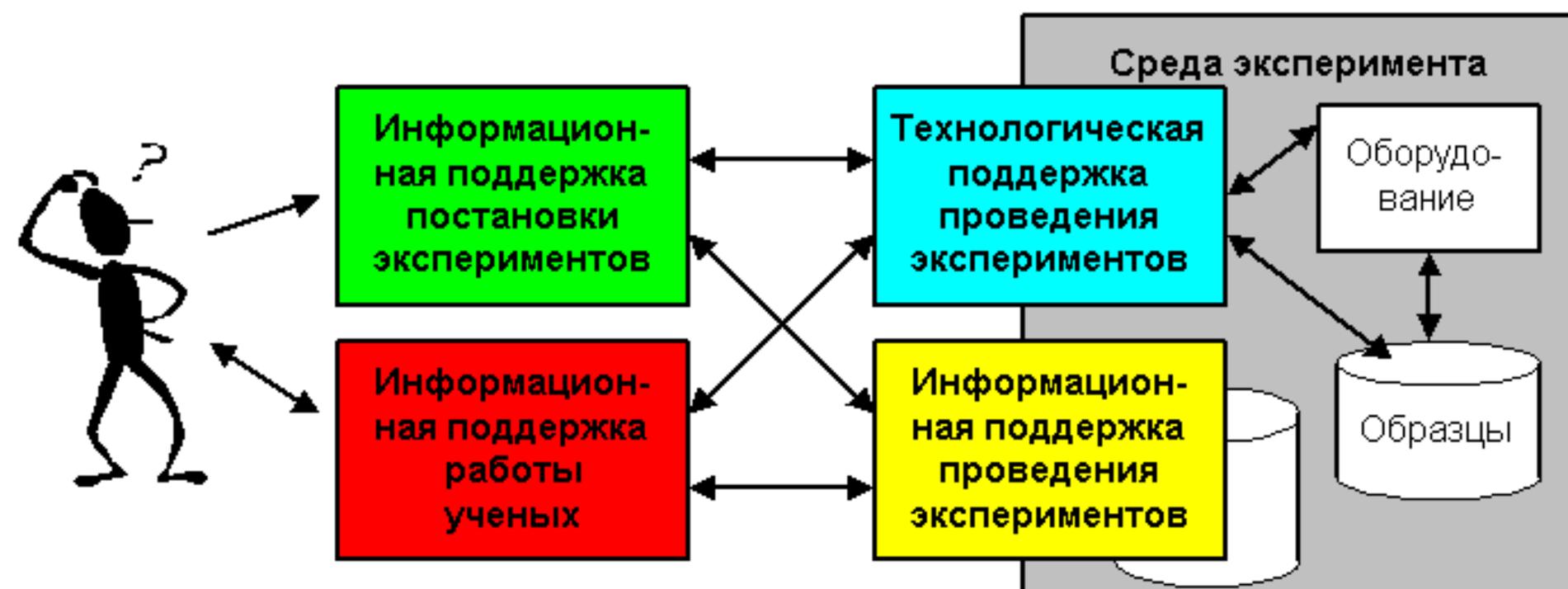
Включает подпроцессы:

- *управления информацией* (включая ее поиск и обработку),
- *постановки экспериментов* и моделирования,
- *научного сотрудничества*, обмена информацией.

1. Процесс научного сотрудничества:



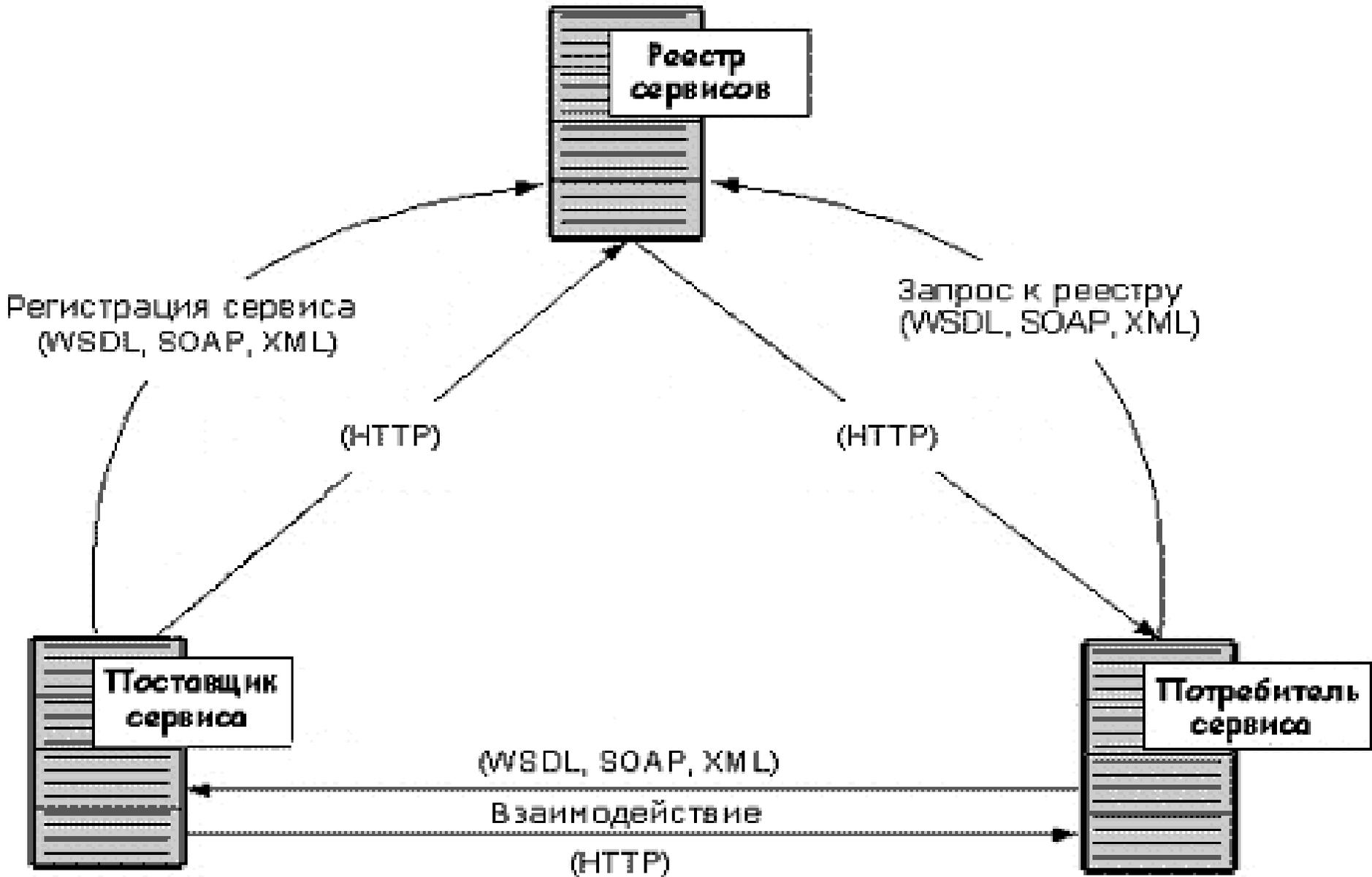
2. Процесс подготовки и проведение экспериментов:



1. Технологическая поддержка экспериментов (на стороне удаленного оборудования)

- Конфигурирование оборудования. Установление интерфейсов;
- Обеспечение надежности интерфейсов с оборудованием;
- Резервирование оборудования для проведения эксперимента;

Средства работы с сервисами в Интернет



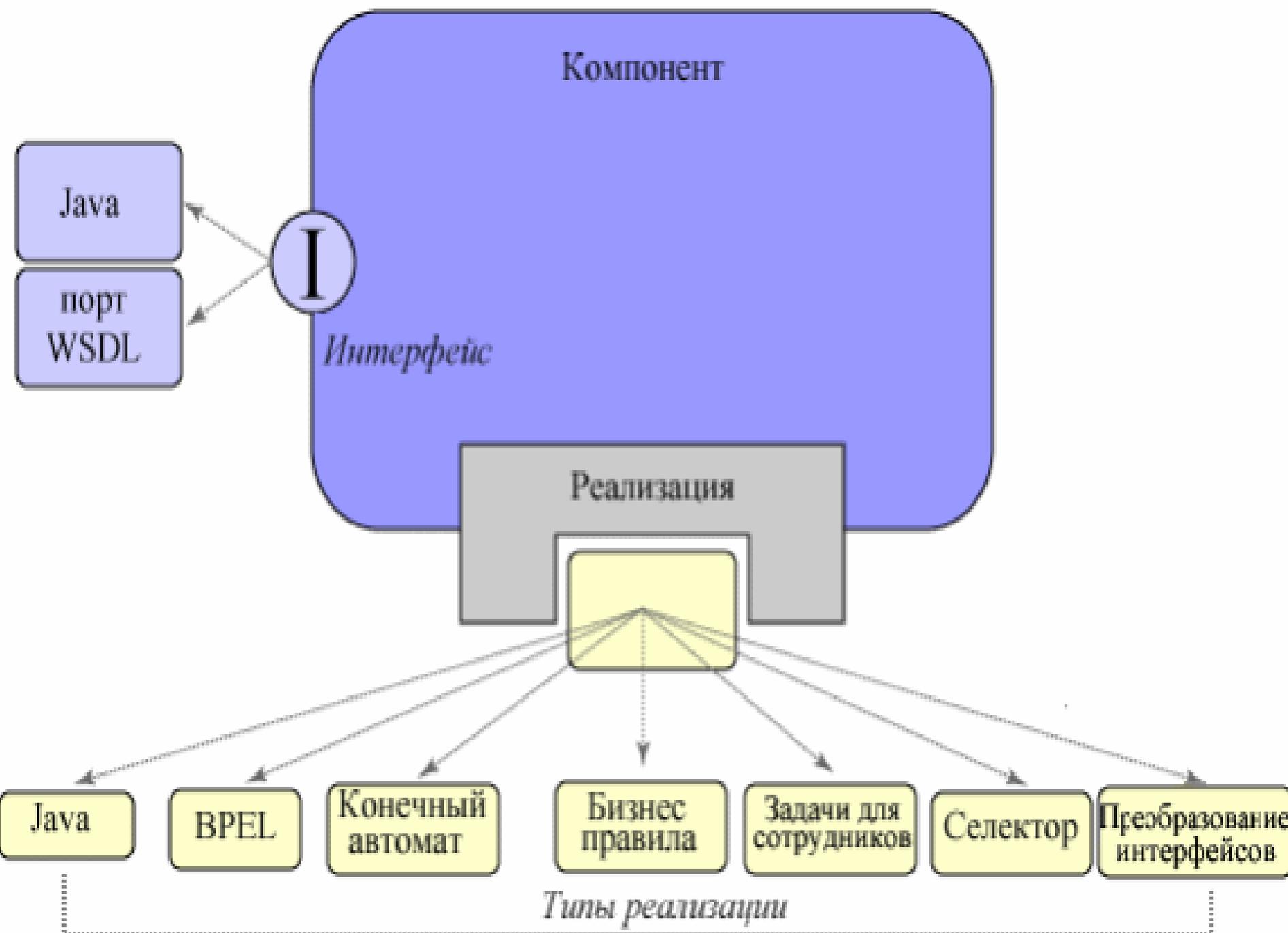
Использование сервисов Интернет

Сервисно-компонентная архитектура (SCA)

SCA обеспечивает доступ к готовым сервисным компонентам системы IBM **WebSphere** Integration Developer (на платформе Eclipse) могут быть упакованы в модуль для выполнения сервиса в WebSphere.

Process Server – эквивалентен EAR-файлу J2EE. Подмодули **J2EE** упаковываются с SCA, запускают сервис и передают данные. **SCA** модуль обращается к **Enterprise JavaBean**. Обмен данными проводится через **SDO (Service Data Objects)**,

В **SCA** сервисы могут собираться в различные образования (**хореографии**). Они и SOA создают новые сервисы, задают их комбинации и конфигурации.



Компонент

I

Интерфейс

Реализация

Бизнес правила

Задачи для сотрудников

Селектор

Преобразование интерфейсов

Типы реализации

Веб-сервисы и ASP.NET

Веб-системы создаются и в архитектуре **ASP.NET**:

- для вызова веб-сервиса используется серверная технология **AJAX**;
- обмен данными между клиентом и сервером осуществляется в формате **JSON (POST HTTP и GET HTTP)**.

На платформе **ASP.NET** :

- создается файл с расширением **ASMX** и с атрибутом **WebService** (точка входа);
- в описании класса задаются открытые методы с атрибутом **WebMethod**;
- функциональность Веб-сервиса задает **ASMX-файл** с передкомпилированными сборками.

Система Jasp для создания Веб-систем

Jasp - это средства для разработки Веб-систем в Microsoft.Net.

В этой платформе обеспечено:

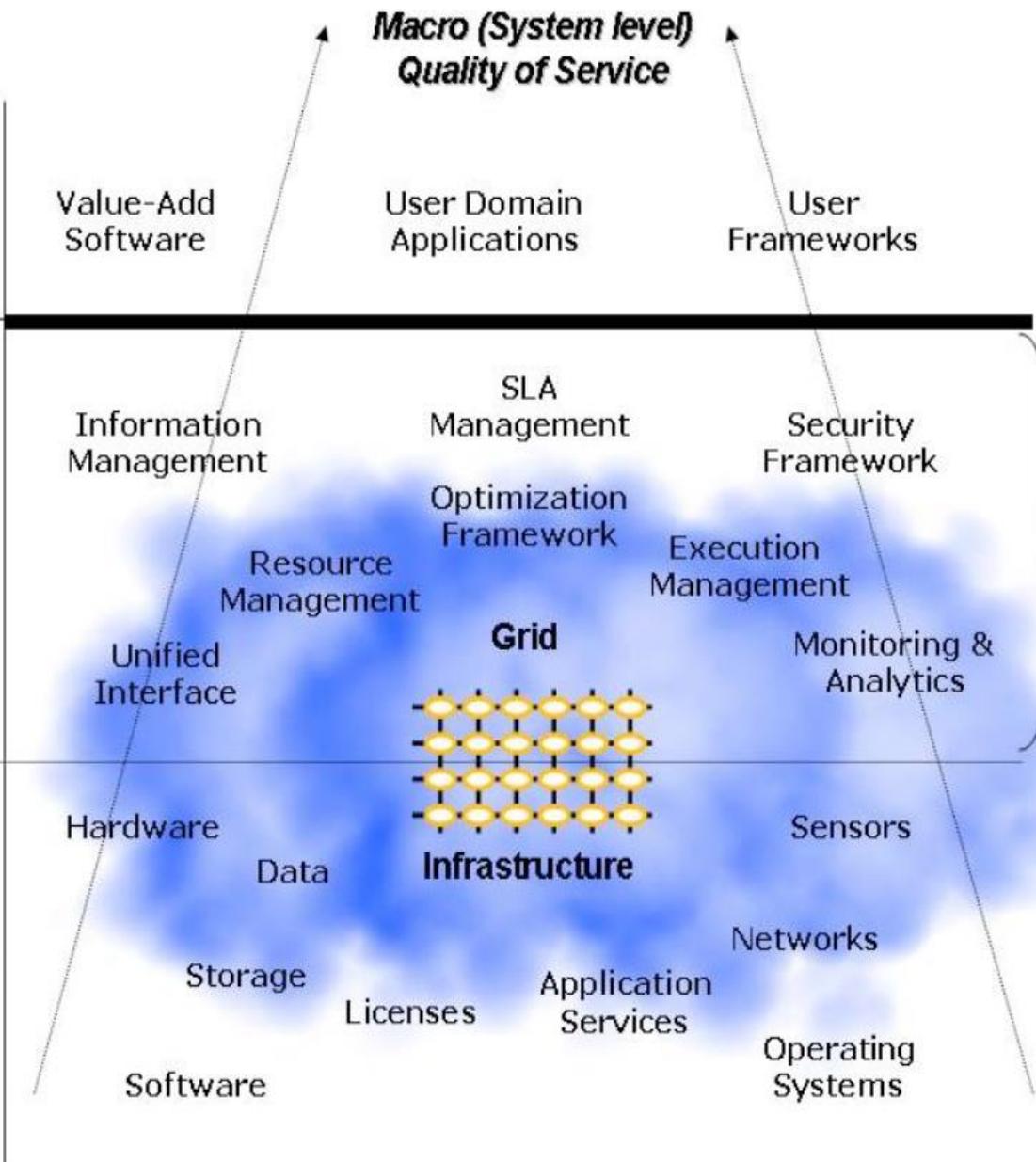
- повторное использование компонент;
- наследование элементов в серверной и клиентской частях;
- **JavaScript, CSS** и разметки;
- автоматическую загрузку ресурсов (синхронно или асинхронно);
- работа с JavaScript-библиотеками: jQuery, ExtJS, Prototype и т.п.

Jasp определяет следующие ресурсы: **CSS, JavaScript – HTML** и бинарные ресурсы. Группирует все CSS и JavaScript файлы в один CSS, или JavaScript файл и добавляет специальные ссылки в HTML разметку для загрузки браузером.

- Business value
- User/Usability focused

- Services oriented
- Virtualizations
- Standards based
- Lower variability

- High variability
- Locally managed
- Locally customized and optimized

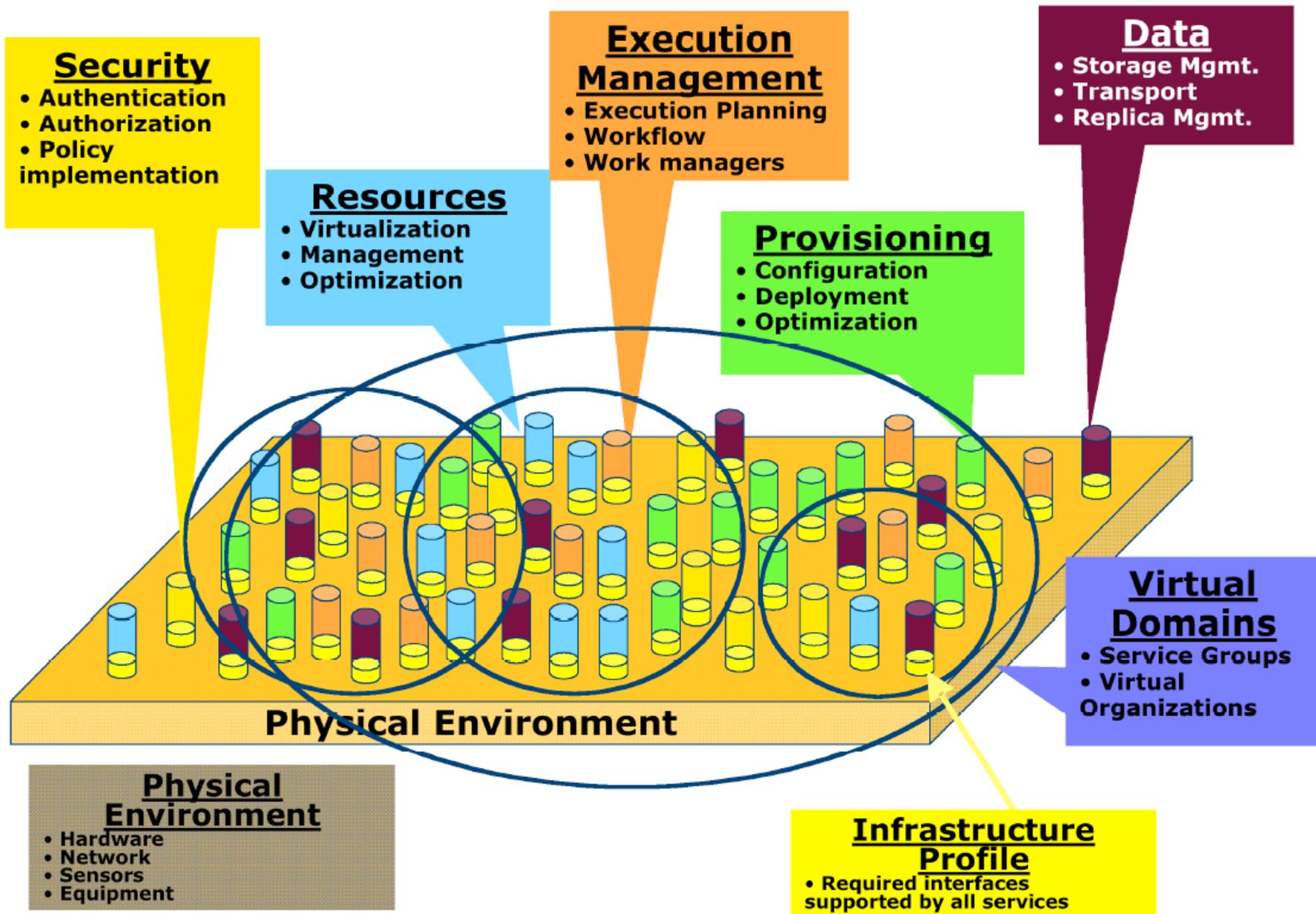


O
G
S
A

F
O
C
U
S

O
G
S
A

R
e
l
e
v
a
n
c
e



Различие между Grid-сервисами и Web-сервисами. Виртуализация

Web-сервисы - не имеют состояния. Срок жизни определяется поставщиком услуги. Существуют независимо от потребителя.

Grid-сервисы

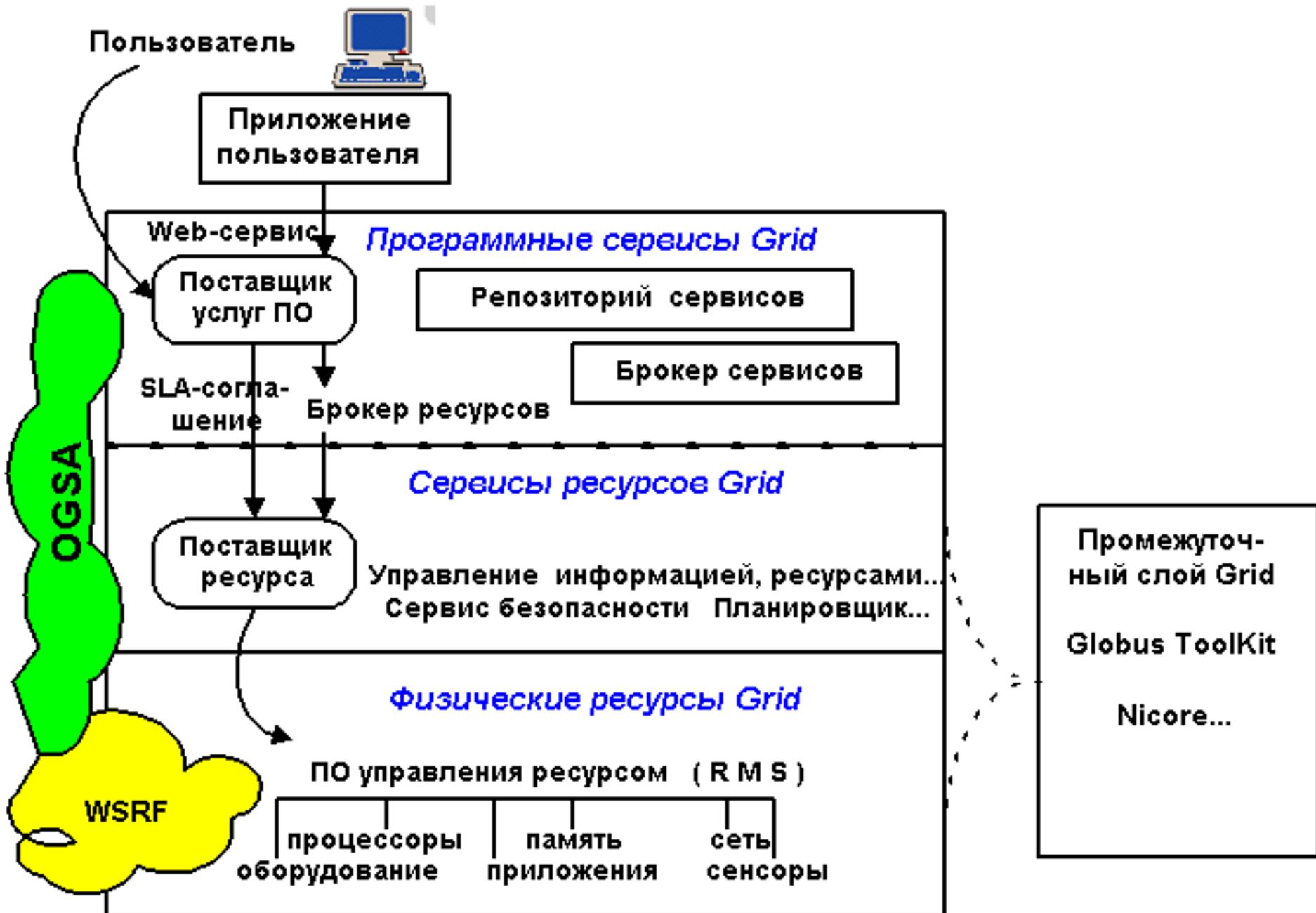
- имеют **состояние** и **срок жизни**, ограниченный *потребностями «потребителя»*,
- «рождаются» на серверах в Grid-сети для решения **конкретной задачи** и «исчезают» по ее завершении.
- представляют «свой» сервер как **фабрику**, на которой **по заказу** клиента создается нужный сервис, используемый в заданное время.

Этот подход называется «**фабрика образцов**» (factory).

Сам сервер (=сервер-посредник) услуг не оказывает. **Фабрикует** нужный «образец» услуг, который оформляется как **сетевой сервис (виртуальный)**.

Если возникает необходимость решить какую-то задачу в Grid-сети, под нее создается **виртуальный компьютер**, предоставляющий нужный **физический сервер**. Этот сервер может поддерживать множество подобных **виртуальных вычислительных машин**, создаваемых под требования клиента и уничтожаемых, когда в них отпадает необходимость.

Обобщенная архитектура Grid



4.Подход к моделированию систем и сайтов в Интернет

А.Г. Рыжов, Е.М. Лаврищева,

XX Всероссийская конференция «Научный сервис в сети Интернет» - 2018

При разработке Веб-систем и приложений используются следующие проектные решения по архитектуре приложения в зависимости от уровня нагрузки и логики Веб-систем.

Клиент, приложение и база данных находятся на одном сервере. Это решение подходит для простого Веб-приложения и имеет плюсы в виде несложной настройки и минимальных затрат на оборудование.

Минусы этого решения:

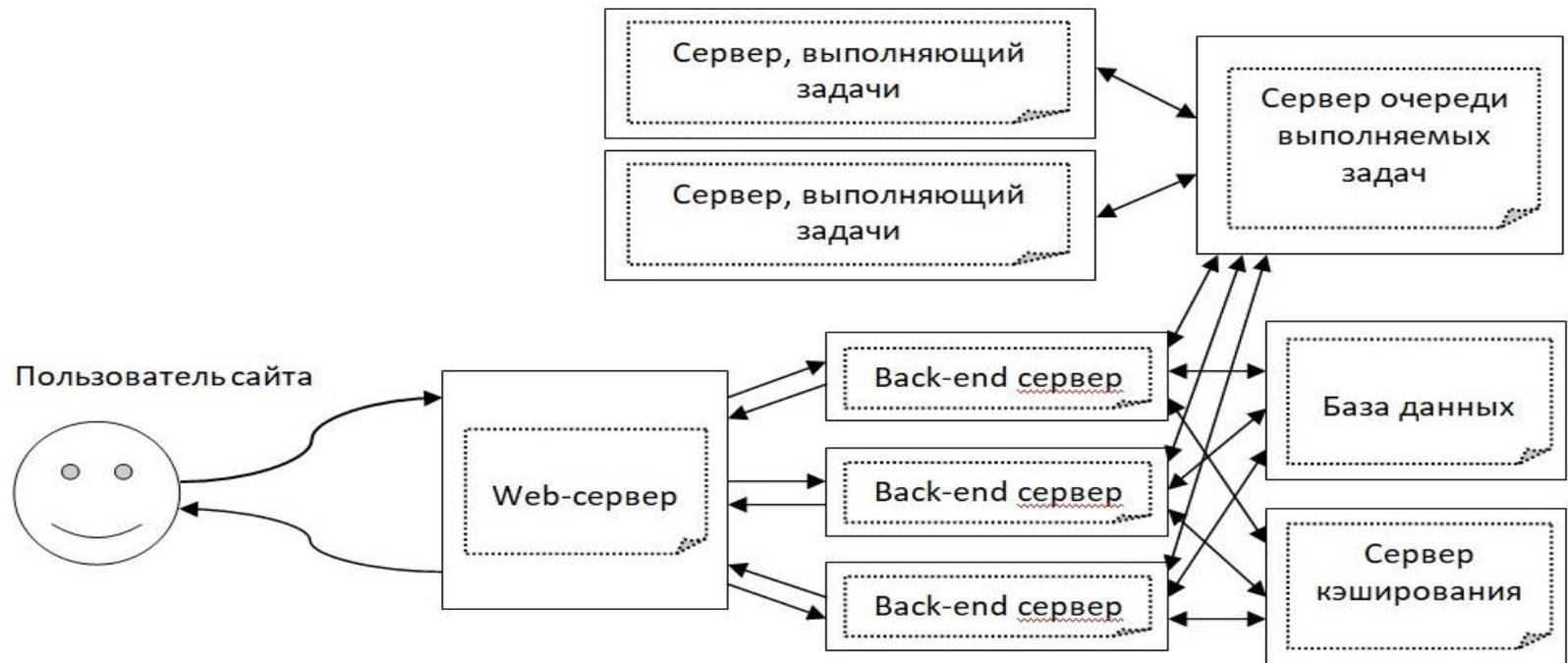
- использование одних и тех же серверных ресурсов снижает производительность и надежность этой системы.**
- сложности при горизонтальном масштабировании.**

Важным узлом Веб-системы является база данных (БД). Каждый запрос от пользователя к Веб-системе может вызывать до сотни и более запросов к БД. Если поместить БД на отдельный сервер, то повышается производительность и снижается негативное влияние на остальные компоненты архитектуры Веб –системы и сайта (производительность, безопасность и надежность). Используется **вертикальное и горизонтальное масштабирование.**

Горизонтальное масштабирование системы управления данными является достаточно сложной задачей. Традиционные SQL-ориентированные СУБД изначально создавались для работы на одной машине и потому плохо приспособлены для работы в кластере или облаке, что привело к разработке новых распределённых систем и подходов, решающих проблему горизонтальной масштабируемости

Сервер очереди задач обработки данных позволяет выполнять задачи асинхронно, передавая их на выполнение вспомогательным серверам и не замедляя работу Веб-системы. Количество вспомогательных серверов следует увеличивать, когда среднее количество задач в очереди будет постепенно расти.

Архитектурно это выглядит следующим образом:



Эффективность Веб-системы

Система должна решать поставленные задачи и хорошо выполнять свои функции в различных условиях и иметь такие характеристики, как надежность, безопасность, производительность, увеличение нагрузки и т.д.

Гибкость системы. Хорошая архитектура позволяет откладывать принятие ключевых решений при разработке.

Расширяемость системы. Возможность добавлять в систему новые функции и функции, не нарушая ее основной структуры с возможностью расширить/изменить поведение системы без изменения/переписывания уже существующих частей системы.

Масштабируемость процесса разработки. Возможность сократить срок разработки за счёт добавления к проекту новых людей. Архитектура должна позволять распараллеливать процесс разработки.

Тестируемость. Код, который легче тестировать, будет содержать меньше ошибок и надежнее работать. Требование «хорошей тестируемости» автоматически приводит к хорошему дизайну и качеству продукта.

Существует целая методология разработки программ на основе тестов, которая так и называется — Разработка через тестирование.

Возможность повторного использования. Систему желательно проектировать так, чтобы ее

Сопровождаемость.

Обеспечение качества созданной Веб-системы

Качество - это совокупность свойств (показателей качества) ПО сайта, которые обеспечивают его способность удовлетворять потребности заказчика, в соответствии с его назначением.

Качество определено в ряде ГОСТ стандартов с помощью модели качества и его показателей.

Стандарт ГОСТ ISO/IEC 12207 определяет основные процессы ЖЦ разработки систем и сайтов и включает процессы, которые регламентируют планирование, управление качеством и оценку затрат на проект. На этапах ЖЦ проводится анализ качества ПО, ориентированные на:

- достижение качества ПО в соответствии с требованиями и критериями;
- верификацию и валидацию промежуточных результатов ПО на этапах ЖЦ и измерение степени достижения отдельных его показателей;
- тестирование готовой ПС, сбор данных об ошибках, дефектах, отказах и др. в системе, а также оценивание надежности по соответствующим моделям надежности.

- Ekaterina M. Lavrischeva. Ontology of Domains. Ontological Description
Software Engineering Domain—The Standard Life Cycle, Journal of Software
Engineering and Applications, 2016, 8, p.1-15. Published Online July 2015 in SciRes.
<http://www.scirp.org/journal/jseai>

-Lavrischeva Ekaterina. Ontological Approach to the Formal Specification
of the Standard Life Cycle, Conference "Science and Information Conference-2015",
July 28-30, London, UK, www.conference.thesai.org.- p.965-972.

Благодарю за внимание