

ЭТАЛОННАЯ МОДЕЛЬ ОКРУЖЕНИЙ ОТКРЫТЫХ СИСТЕМ

Open System Environment - Basic Reference Model (OSE RM)

СОДЕРЖАНИЕ

1. КОНЦЕПЦИЯ ОТКРЫТЫХ СИСТЕМ (ОС)
2. Методологическая основа концепции ОС
3. Эталонная модель OSE RM
 - Три типа сущностей OSE RM
 - ✓ Сущность "прикладное программное обеспечение"
 - ✓ Сущность "прикладная платформа"
 - ✓ Внешнее окружение
 - Два типа интерфейсов OSE RM: API и EEI
 - Структуризация интерфейсов
 - OSE RM (ISO/IEC 14252)

КОНЦЕПЦИЯ ОТКРЫТЫХ СИСТЕМ

Курс лекций В.А. Сухомлина

"ВВЕДЕНИЕ В АНАЛИЗ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ"

<http://old.master.cmc.msu.ru/lectures/AnalyzeIT/index.htm>

- **переносимость и переиспользуемость** программного обеспечения, данных и опыта людей;
- **интероперабельность**, т.е. возможность взаимодействия компонентов распределенной системы посредством обмена информацией и ее совместного использования;
- **масштабируемость** как свойство сохранения работоспособности системы ИТ в условиях варьирования значений параметров, определяющих технические и ресурсные характеристики системы и/или поддерживающей среды.

Методологическую основу концепции открытых систем (ОС) составляют:

- концептуальный базис и принципы построения ОС
- эталонная модель окружений ОС (RM OSE)
- эталонная модель взаимосвязи открытых систем (RM OSI)
 - предмет нашего курса
- аппарат профилирования ИТ, предназначенный для конструирования ОС в пространстве стандартизованных решений
- концепция тестирования конформности систем ИТ исходным стандартам и профилям
- таксономия профилей

Основные документы, определяющие методологическую основу концепции открытых систем (1):

- 1) Технический отчет ISO/IEC TR 10000 Framework and taxonomy of International Standardized Profiles (**Основы и таксономия международных стандартизованных профилей**) в трех частях [1, 2, 3], включая:
 - Часть 1: General Principles and Documentation Framework (**Общие принципы и основы документирования**)
 - Часть 2: Principles and Taxonomy for OSI Profiles (**Принципы и таксономия профилей взаимосвязи открытых систем**)
 - Часть 3: Principles and Taxonomy for Open System Environment Profiles (**Принципы и таксономия профилей окружений открытых систем**)

Основные документы, определяющие методологическую основу концепции открытых систем (2):

- 2) **Эталонная модель окружения (среды) открытых систем (*RM OSE*)** - ISO/IEC DTR 14252, Portable Operating System Interface for Computer Environments - POSIX. (IEEE, P1003.0, Draft Guide to the POSIX **O**pen **S**ystem **E**nvironment)
- 3) **Эталонная модель взаимосвязи открытых систем (*RM OSI*)** - ISO 7498:1996, Information processing systems - **O**pen **S**ystems **I**nterconnection - Basic **R**eference **M**odel [ITU-T Rec. X.200]

Открытая система по IEEE POSIX 1003.0

● Открытые системы – это системы

- в которых реализован "исчерпывающий и согласованный набор международных стандартов информационных технологий и профилей функциональных стандартов, которые специфицируют интерфейсы, сервисы и поддерживаемые форматы данных,
- чтобы обеспечить интероперабельность и мобильность приложений, данных и персонала" (определение IEEE)

● Ключевые слова

- “специфицируют **интерфейсы, сервисы** и поддерживаемые **форматы данных**”

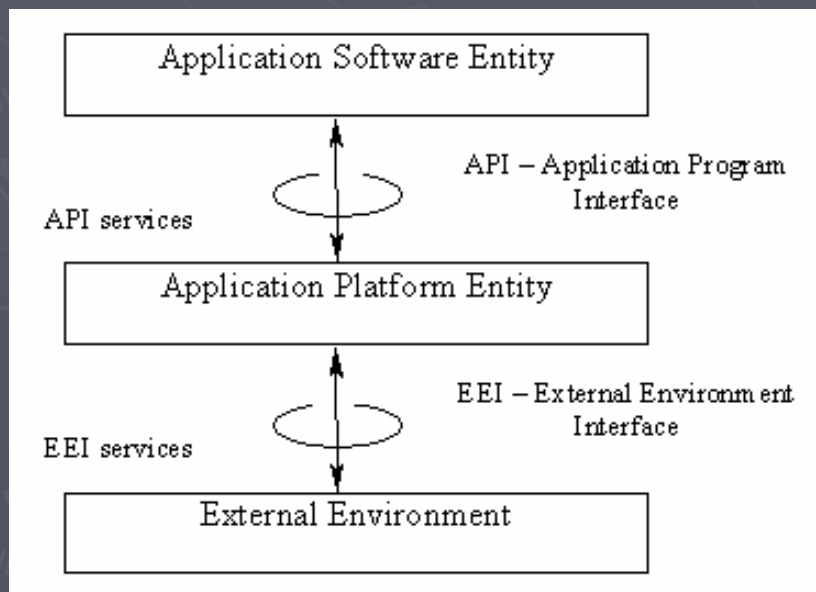
Эталонная модель OSE RM

- OSE RM рассматривает **Информационную систему** как **черный ящик**
- **Взаимодействие** с черным ящиком
 - **стандартизовано**
 - осуществляется только **через интерфейсы**
 - ✓ под интерфейсами понимаются границы, разделяющие между собой некоторые функциональные сущности черного ящика
- Через интерфейсы система (платформа)
 - **предоставляет сервисы** пользователям (приложениям) и
 - **использует сервисы**, связанные с сущностями внешнего окружения.
- **Центральным понятием OSE RM** является **понятие окружения** открытых систем (Open System Environment - OSE), под которым понимается
 - полный **набор интерфейсов, сервисов, форматов**, а также пользовательских аспектов,
 - **обеспечивающих интероперабельность** и/или **переносимость** программ, данных, людей на основе использования базовых стандартов и профилей ИТ

Три типа сущностей OSE RM

1. Прикладное программное обеспечение (Application Software Entities)
2. Прикладная платформа (Application Software Entities)
3. Внешнее окружение (External Environment Entities)

• Два типа интерфейсов: API и EEI.



Два типа интерфейсов OSE RM: API и EEI

- **стандарты прикладных программных интерфейсов** (Application Program Interface (API) Standards);
 - специфицирует взаимодействие прикладного программного обеспечения с компьютерной системой (прикладной платформой)
 - в основном предназначены для обеспечения переносимости приложений
- **стандарты внешнего окружения** (External Environment Interface (EEI) Standards).
 - определяет взаимодействие информационной системы с ее внешним окружением
 - позволяют решать проблемы
 - ✓ интероперабельности систем,
 - ✓ переиспользуемости программного обеспечения и
 - ✓ переносимости данных.

Сущность "прикладное программное обеспечение"

Состоит из компонент следующих типов:

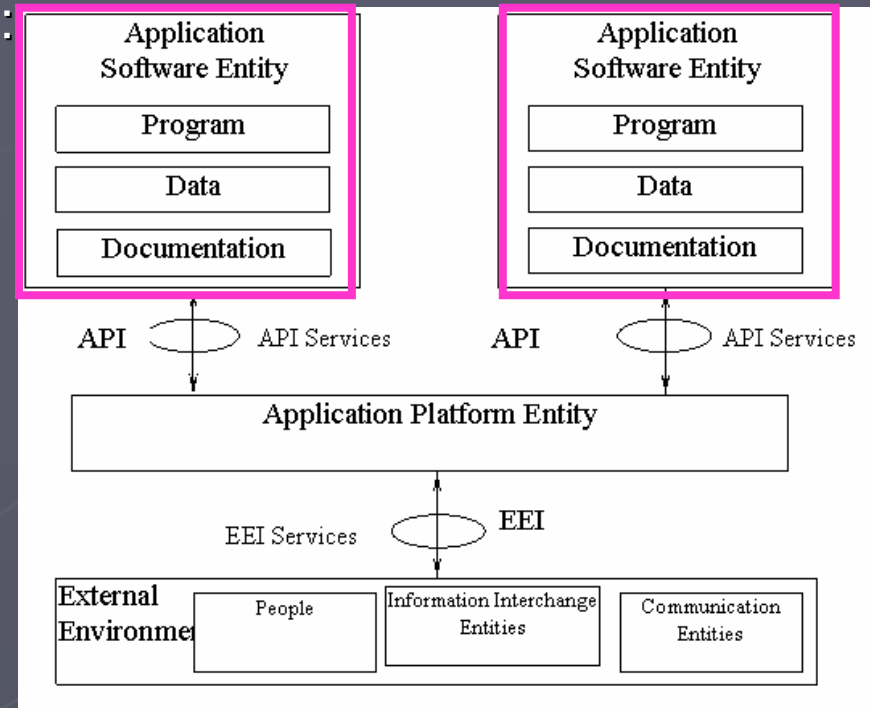
- **Программ** (исходный код, файлы команд или сценариев и т.д.).
- **Данных** (данные пользователя, параметры приложения, параметры экрана и т.д.).
- **Документации** (online)

Может быть разделена на две части:

- **неизменяемую часть** при переносе с одной платформы на другую;
- **изменяемую часть** при переносе с одной платформы на другую.

Для повышения степени переносимости необходимо:

- **использовать** при его создании стандарты на **API**
- **минимизировать изменяемую часть**



На одной и той же прикладной платформе могут одновременно выполняться несколько приложений

На рисунке два приложения на одной платформе

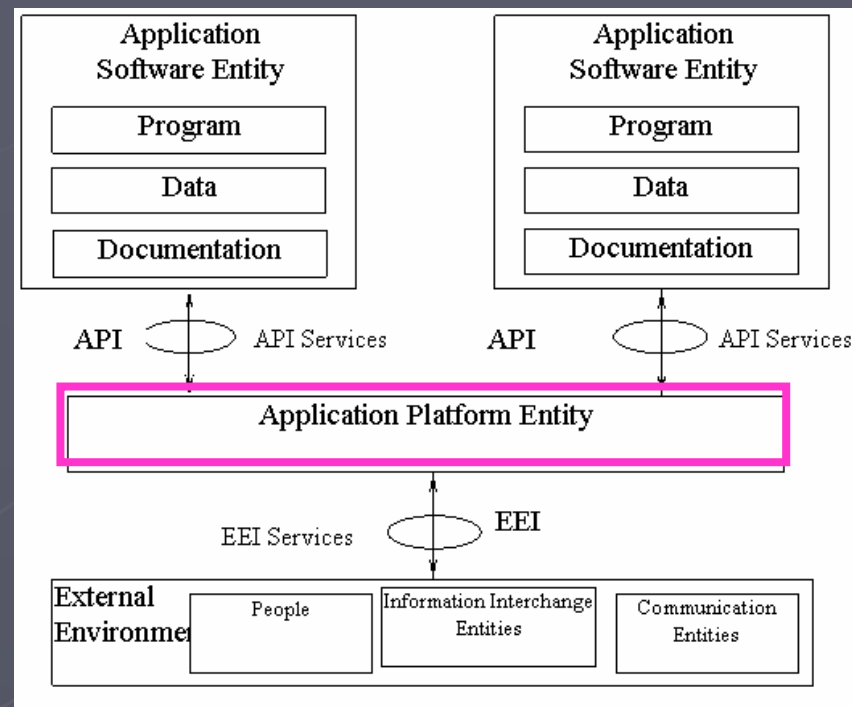
Сущность "прикладная платформа"

● Прикладная платформа

- это набор ресурсов
- ресурсы предоставляются сервисам
- сервисы используются приложениями
- все ресурсы платформы доступны исключительно через API

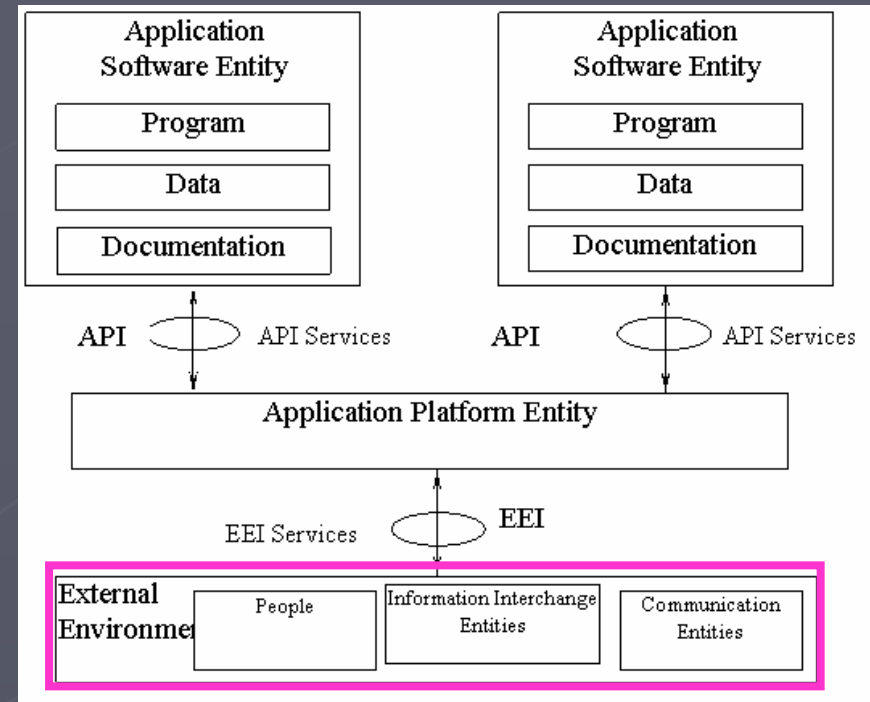
● Понятие прикладной платформы не включает в себя конкретную реализацию сервисов

- Например, платформа может содержать единственный процессор, разделяемый несколькими приложениями, или являться большой распределенной системой



Внешнее окружение

- Внешнее окружение содержит внешние сущности
 - с которыми прикладная платформа обменивается информацией
- Категории сущностей внешнего окружения
 - конечные пользователи (people)
 - информационный обмен
 - коммуникационные



Структуризация интерфейсов

API сервисы:

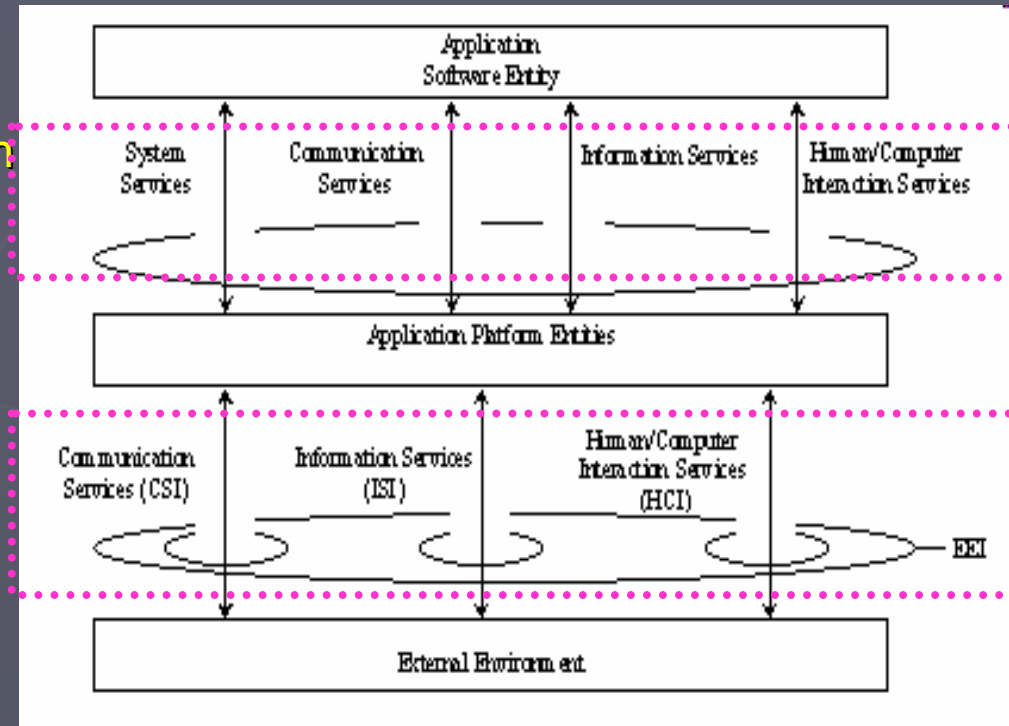
1. системные (**System Services**)
2. коммуникационные (**Communication Services**)
3. информационные (**Information Services**)
4. взаимодействия человека с компьютером (**Human/Computer Interaction Services**)

Пример: Open_Window (x1, y1, x2, y2).

EEL сервисы и соответственно интерфейсы:

1. коммуникационные сервисы (**Communication Services**);
2. информационные сервисы (**Information Services**);
3. сервисы взаимодействия человека с компьютером (**Human/Computer Interaction Services**)

Пример: понятие окна ('window'), как сущности, связанной с некоторой областью на экране монитора

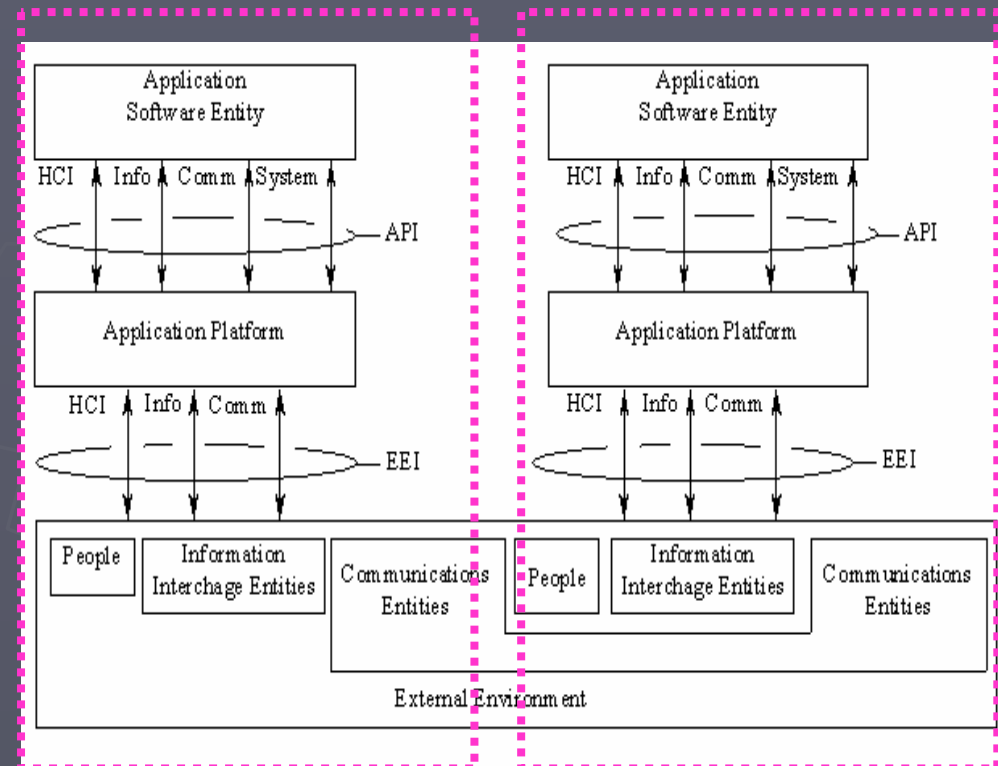


Сервисы API и EEL не имеют взаимно однозначного соответствия, хотя в большинстве случаев имеют похожие названия.

Например, интерфейс сервиса хранения данных может обеспечивать приложению прозрачный доступ к удаленному файлу с помощью сетевых сервисов. В данном случае комплексный сервис хранения данных, предоставляемый API, зависит от коммуникационных сервисов, предоставляемых

OSE RM распределенных систем

- Прикладные программы разных платформ взаимодействуют через API
- API запросы транслируются платформой в соответствующие действия на EEI
- Связь между прикладными платформами устанавливается с помощью внешних сущностей
 - коммуникационными сервисами EEI
- ИТАК: Распределенные в пространстве прикладные платформы взаимодействуют с помощью сервисов (коммуникационных механизмов) интерфейса внешнего окружения (EEI)



Сервисы API и EEI не имеют взаимно однозначного соответствия, хотя в большинстве случаев имеют похожие названия.

Например, интерфейс сервиса хранения данных может обеспечивать приложению прозрачный доступ к удаленному файлу с помощью сетевых сервисов. В данном случае комплексный сервис хранения данных, предоставляемый API, зависит от коммуникационных сервисов, предоставляемых

Способы спецификации API

- В OSE RM рассматриваются следующие **три вида** спецификаций API:
 1. **Средствами некоторого языка программирования** (Programming language API specifications).
 2. **Языково-независимые спецификации сервисов** (Language-independent service specifications).
 3. **Спецификации API языкового связывания** (Language-binding API specifications)
 - ✓ отображения языково-независимых спецификаций в форму доступа к сервису, выраженную средствами конкретного языка программирования (**UML что-ли ? МФ**).
- Примечание: Последние два способа спецификации API считаются наиболее предпочтительными для целей стандартизации.
- От общих архитектурных и понятийных моделей в Руководстве POSIX (эталонная модель OSE RM описана в третьем разделе) осуществляется переход к описанию собственно категорий сервисов, чему посвящен самый большой раздел Руководства, четвертый. При этом каждая категория сервисов описывается по следующей единой схеме: и.т.д. (не наш предмет)

РЕЗЮМЕ: OSE RM (ISO/IEC 14252)

Как бы три уровня

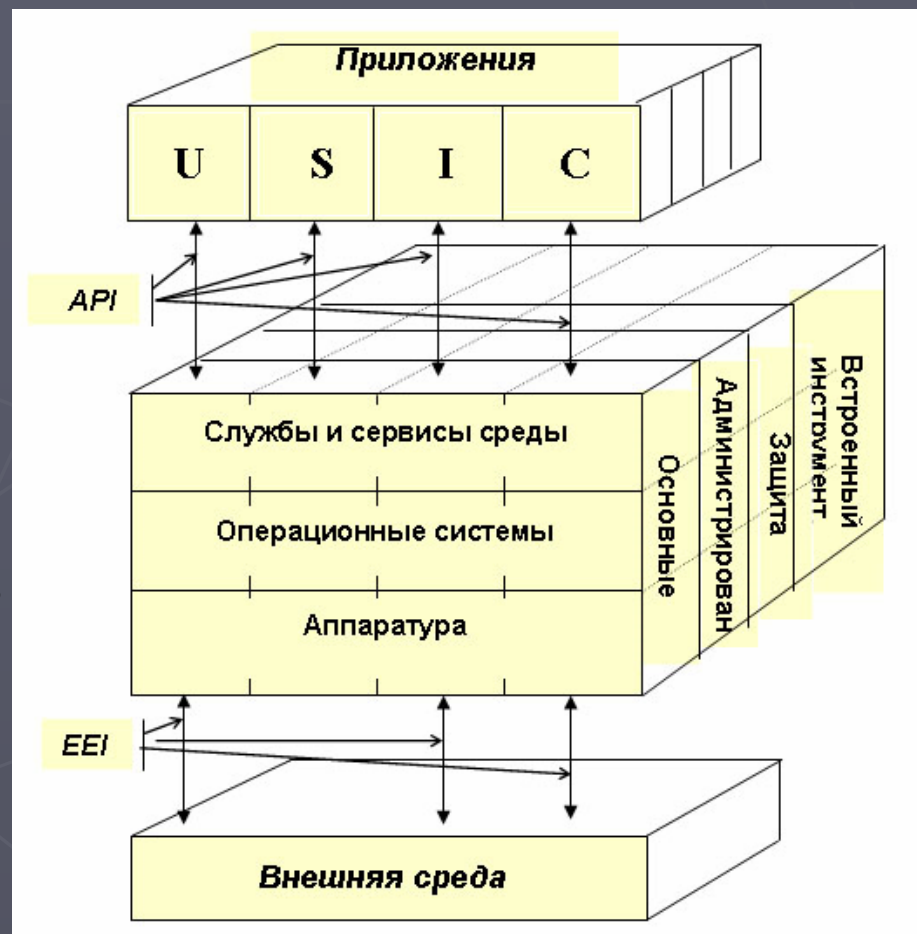
1. **Приложения** (прикладные программные комплексы), реализующие заданные функции ИС
2. **Платформа**, обеспечивает подготовку и выполнение (runtime) приложений
3. **Внешняя среда**

Связаны интерфейсами

- API (application program interface) - интерфейсы прикладного программирования
- EEI (external environment interface) - Интерфейсы внешнего окружения

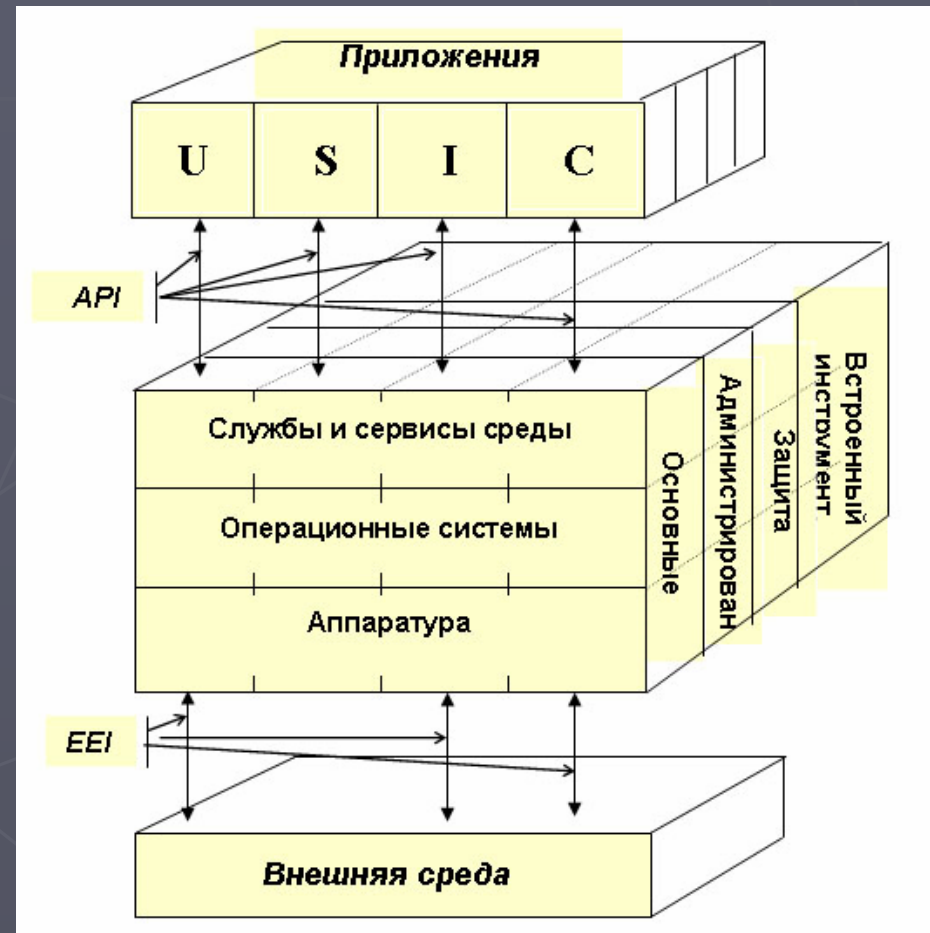
Четыре функциональные группы компонентов

1. **U – user**/пользовательские функции, обслуживающие интерфейс с пользователями;
2. **S – system**/системные функции среды по организации процессов обработки данных
3. **I – information**/функции представления и хранения данных;
4. **C – communication**/коммуникационные функции



OSE RM (ISO/IEC 14252)

- Эти функции могут быть реализованы как приложениями, так и компонентами среды ИС. Их спецификации составляют плоскость основных функций ИС.
- Функции системного и сетевого администрирования распределены между компонентами среды и приложений. Они образуют вторую плоскость концептуальной модели, в которую включаются управление приложениями, управление средствами пользовательского интерфейса, управление базами данных, управление процессами, обеспечиваемое операционными системами, управление коммуникационной сетью или отдельными узлами сети, управление средствами защиты информации.
- Функции средств защиты информации в ИС также распределены между разными компонентами ИС. Часть из них реализуется штатными средствами, встроенными в операционные системы, СУБД, ПО промежуточного слоя (например, в мониторы транзакций), а часть обеспечивается специальными средствами защиты. Поэтому в концептуальную модель введена третья плоскость - функции защиты информации.
- Наконец, четвертую плоскость составляют функции инструментальных средств, встроенных в ИС для поддержки ее эксплуатации и сопровождения.



ЛИТЕРАТУРА

[1]