Семинар 2

ЦЕЛОСТНОСТЬ

И

ДОСТУПНОСТЬ

ДАННЫХ

Понятие ЦЕЛОСТНОСТИ данных

- свойство, гарантирующее точность и полноту информации, а также методов ее обработки

- такое свойство информации, при котором ее содержание и структура (данных) определены уполномоченными лицами и процессами

неизменность — **объектов, содержащих код ПО**

неискаженность — **отсутствие подделки при:**
- передаче данных
- обработке данных

правильность — **отсутствие ошибок в:**
- структуре данных
- содержании данных

Обеспечение ЦЕЛОСТНОСТИ данных

- разграничение доступа к данным, обеспечивающее их целостность;

- криптографические технологии электронной цифровой подписи;

- организация параллельного выполнения транзакций в клиент-серверных СУБД.

Дискреционная модель Кларка-Вильсона (1987г.)

Главная идея «тройки целостности»:
«субъект-операция(транзакция), не нарушающая целостность - объект»

Исходные положения

1. Все множество объектов ***D*** разделяется на объекты ***CDI*** , требующие контроля целостности (*constrained data items*), и объекты ***UDI***, не требующие контроля целостности (*unconstrained data items*)

$$D = CDI \cup UDI, \quad CDI \cap UDI = \varnothing$$

2. На множестве элементарных операций над объектами выделяются совокупности (последовательности), обособляющиеся в логически самостоятельные сущности, называемые процедурами преобразования ***TP*** (*transformation procedures*)
3. Дополнительно вводится особый класс процедур *IVP* над данными, которые обеспечивают проверку целостности контролируемых данных (*integrity verification procedures*)
4. Те процедуры преобразования данных *TP*, применение к результатам которых процедур проверки целостности *IVP* дает положительный результат, называются «*корректно* (*правильно, хорошо*) *сформированными транзакциями*»

Дискреционная модель Кларка-Вильсона (1987г.)

Правила функционирования системы

C1. Множество всех процедур контроля целостности *IVP* должно содержать процедуры контроля целостности любого элемента данных из множества всех *CDI*

C2. Все процедуры преобразования *TP* должны быть хорошо сформированными транзакциями, т.е. не нарушать целостности данных, и применяться только по отношению к списку элементов (объектов) *CDI*, устанавливаемых администратором системы

E1. Система должна контролировать допустимость применения *TP* к элементам *CDI* в соответствии со списками, указанными в правиле C2

E2. Система должна поддерживать список разрешенных конкретным пользователям процедур преобразования *TP* с указанием допустимого для каждой *TP* и данного пользователя набора обрабатываемых элементов *CDI* (т.е. тройки «субъект-*TP*-объект *CDI*»)

Дискреционная модель Кларка-Вильсона (1987г.)

Правила функционирования системы

C3. Список, определенный правилом C2, должен отвечать требованию разграничения функциональных обязанностей (в т.ч. совм. выполнения)

E3. Система должна аутентифицировать всех пользователей, пытающихся выполнить какую-либо процедуру преобразования *TP*

C4. Каждая *TP* должна записывать в журнал регистрации информацию, достаточную для восстановления полной картины каждого применения этой *TP*. Журнал регистрации – это специальный элемент *CDI*, предназначенный только для добавления в него информации

C5. Специальные *TP* могут корректно обрабатывать *UDI*, превращая их в *CDI*

E4. Только специально уполномоченный субъект (пользователь) может изменять списки, определенные в правилах C2 и E2. Этот субъект не имеет права выполнять какие-либо действия, если он уполномочен изменять регламентирующие эти действия списки

Мандатная модель целостности

Система защиты – совокупность:
- множества субъектов *S*
- множества объектов *O*
- множества операций над объектами доступа *R* (два элемента - *read* и *write*)
- решетки уровней безопасности Λ субъектов и объектов (решетка уровней целостности данных)
- функции f, отображающей элементы множеств *S* и *O* в Λ
- множества состояний системы *V*, которое определяется множеством упорядоченных пар (*F,A*)
- начального состояния v_0
- набора запросов *Q*
- функции переходов *T:* $(V \times Q) \rightarrow V$, которая переводит систему из одного состояния в другое при выполнении запросов субъектов на доступ к объектам

Мандатная модель целостности Биба

Критерий безопасности (обеспечения целостности):
- недопустимы потоки «снизу вверх», т.к. могут нарушить целостность объектов более высокого уровня безопасности:

- запись данных субъектом $s \in S$ в объект $o \in O$ с более высоким уровнем безопасности – $\mathcal{F}(s) < \mathcal{F}(o)$ (*no write up - NWU)* – нельзя писать вверх, т.к. в результате может произойти нарушение целостности («загрязнение») объекта

- чтение данных субъектом $s \in S$ из объекта $o \in O$ с более низким уровнем безопасности – $\mathcal{F}(o) < \mathcal{F}(s)$ (*no read down - NRD*) – нельзя читать вниз, т.к. в результате может произойти нарушение целостности («загрязнение») субъекта

Сравним модели Биба и Белла-ЛаПадулы

1. Состояние называется безопасным по чтению (или *просто безопасным*) тогда и только тогда, когда для каждого субъекта, осуществляющего в этом состоянии доступ чтения к объекту, уровень безопасности этого субъекта доминирует над уровнем безопасности этого объекта:

$$\forall s \in S,\ \forall o \in O,\ read \in A[s,o] \rightarrow f_L(s) \geq f_L(o)$$

2. Состояние называется безопасным по записи (или **-безопасным*) тогда и только тогда, когда для каждого субъекта, осуществляющего в этом состоянии доступ записи к объекту, уровень безопасности объекта доминирует над уровнем безопасности этого субъекта:

$$\forall s \in S,\ \forall o \in O,\ write \in A[s,o] \rightarrow f_L(o) \geq f_L(s)$$

3. Состояние безопасно тогда и только тогда, когда оно безопасно *и по чтению, и по записи*

Мандатная модель Биба (сер.70-х годов, Кен Биба)

Модель Биба - инверсия модели Белла-ЛаПадулы

Разновидности модели Биба

Модель с понижением уровня субъекта

Субъекты могут читать любые объекты

но, т.к. в результате они могут быть загрязнены, то после завершения операции чтения, уровень целостности субъектов должен быть понижен до уровня целостности прочитанного объекта

$$\mathcal{F}(o) < \mathcal{F}(s)$$

$$\mathcal{F}^*(s) \equiv \mathcal{F}(o)$$

Мандатная модель Биба (сер.70-х годов, Кен Биба)

Модель с понижением уровня объекта

Субъекты могут писать в любые объекты

но, т.к. в результате объект может быть загрязнен, то после завершения операции записи, уровень целостности измененного объекта должен быть понижен до уровня целостности изменяющего субъекта

$$\mathcal{F}(s) < \mathcal{F}(o)$$

$$\mathcal{F}^*(o) \equiv \mathcal{F}(s)$$

Главный недостаток - «деградация» системы

Мандатная модель Биба (сер.70-х годов, Кен Биба)

Возможности объединения мандатной модели обеспечения конфиденциальности Белла-ЛаПадулы с мандатной моделью обеспечения целостности Биба

1. На основе двух различных решеток в одной КС

- решетка $\Lambda_к$ уровней конфиденциальности и функция отображения на нее субъектов и объектов доступа $\mathcal{F}_к(x) = l_к \in \Lambda_к$, $x \in S \cup O$

- решетка $\Lambda_ц$ уровней целостности и функция отображения на нее субъектов и объектов доступа $\mathcal{F}_ц(x) = l_ц \in \Lambda_ц$, $x \in S \cup O$

- принятие решения на доступ одновременно по правилам *NRU* и *NWD* по функции $\mathcal{F}_к(x)$ и правилам *NRD* и *NWU* по функции $\mathcal{F}_ц(x)$

но м.б. противоречия и тупики

Мандатная модель Биба (сер.70-х годов, Кен Биба)

Возможности объединения мандатной модели обеспечения конфиденциальности Белла-ЛаПадулы с мандатной моделью обеспечения целостности Биба

2. На основе одной общей решетки

- операции *read* и *write* возможны только в пределах одного уровня безопасности $F(s) = F(o)$ - полностью изолированная по уровням безопасности система (т.н. «*равное чтение*» и «*равная запись*»)

3. На основе одной общей решетки, со спец. отображением

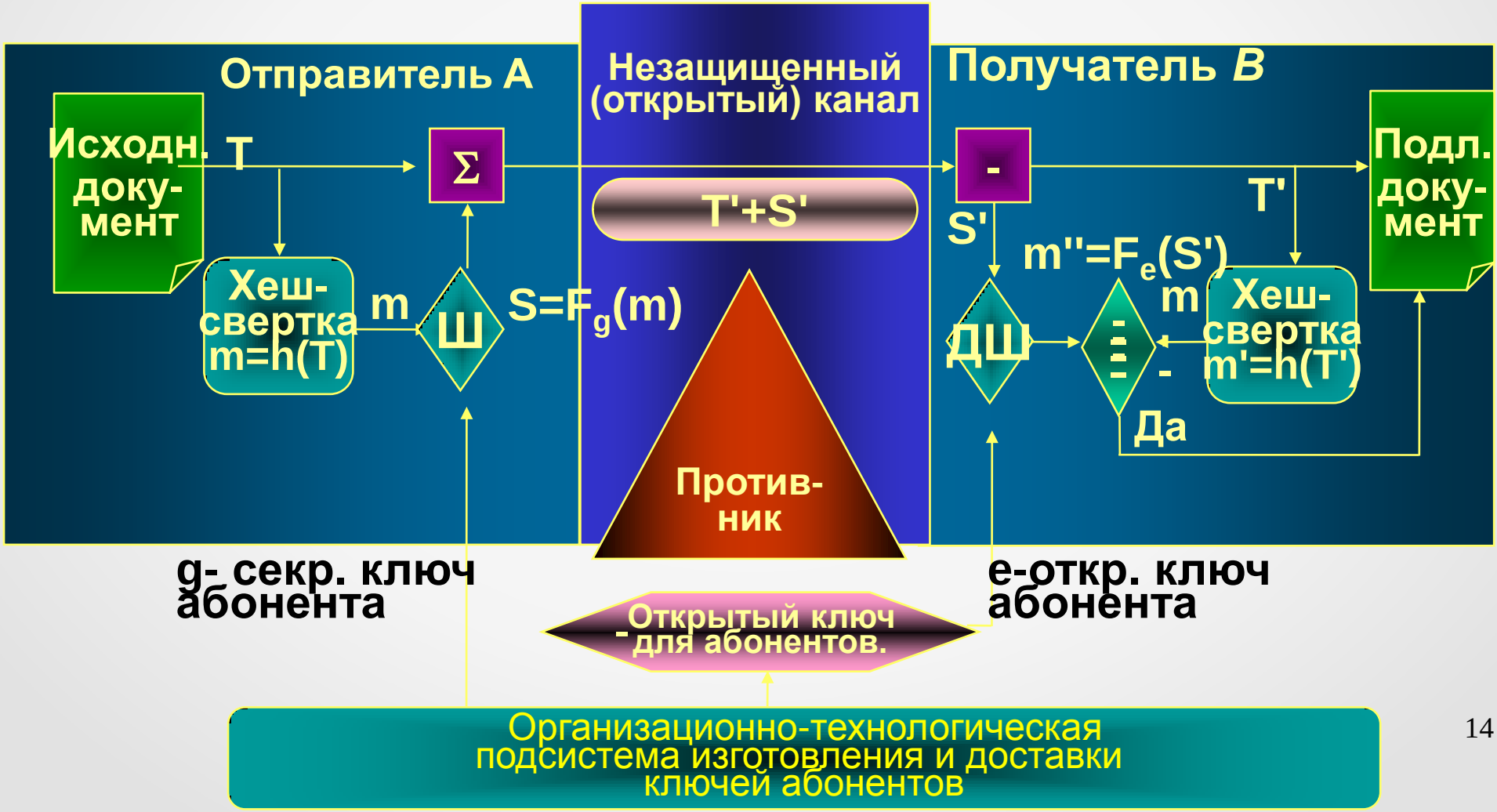
- субъекты и объекты с высокими требованиями *целостности* располагаются на *нижнем* уровнем иерархии решетки (сист.ПО и прогр-ст)

- субъекты и объекты с высокими требованиями *конфиденциальности* располагаются на *самом высоком* уровне иерархии решетки (секр. данные и доверенные пользователи)

Обобщенная структурная схема системы ЭЦП

Система ЭЦП включает два этапа:

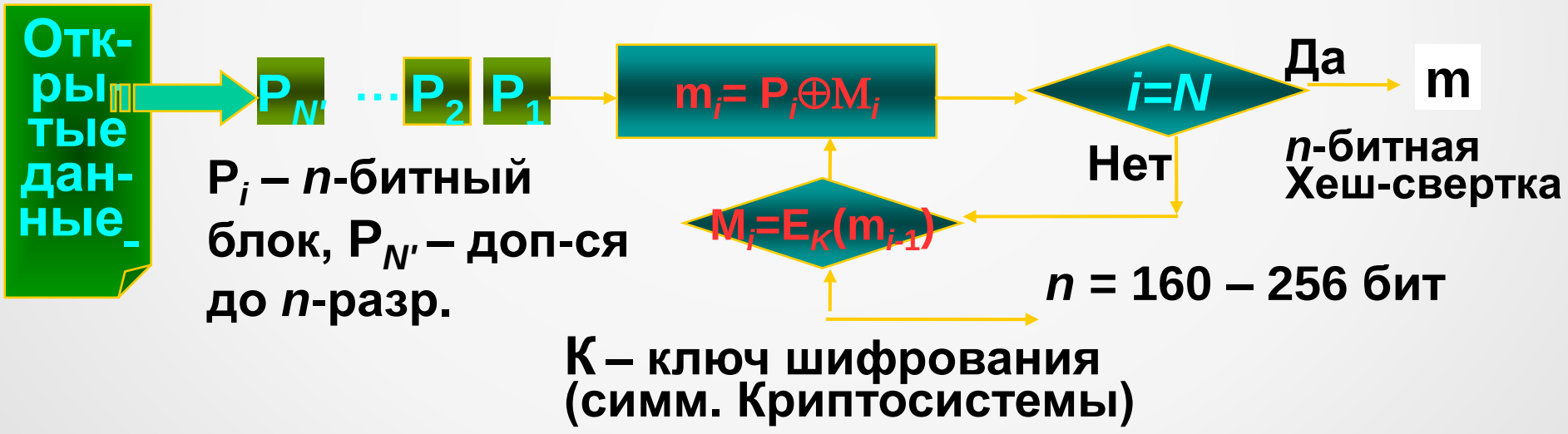
- **процедуру постановки подписи**
- **процедуру проверки подписи**



хеш-функции

Как правило, хеш-функции строят путем итерационных процедур на основе одношаговых сжимающих функций

Обобщенная схема ключевой хеш-функции



Инфраструктура открытых ключей

Абонент, получивший сообщение, должен быть уверен, что открытый ключ, с помощью которого расшифровывается сообщение или проверяется ЭЦП, действительно принадлежит объявленному отправителю

Сертификат ключа

- набор данных, заверенный ЭЦП центра сертификации (удостоверяющего центра) и включающий открытый ключ и список атрибутов, относящихся к абоненту ключа (имя абонента, название центра сертификации, номер сертификата, время действия сертификата, предназначение ключа (шифрование, ЭЦП))
- проверив ЭЦП сертификата по известному открытому ключу сертификационного центра, можно убедиться, что находящийся в нем открытый ключ действительно принадлежит обозначенному пользователю

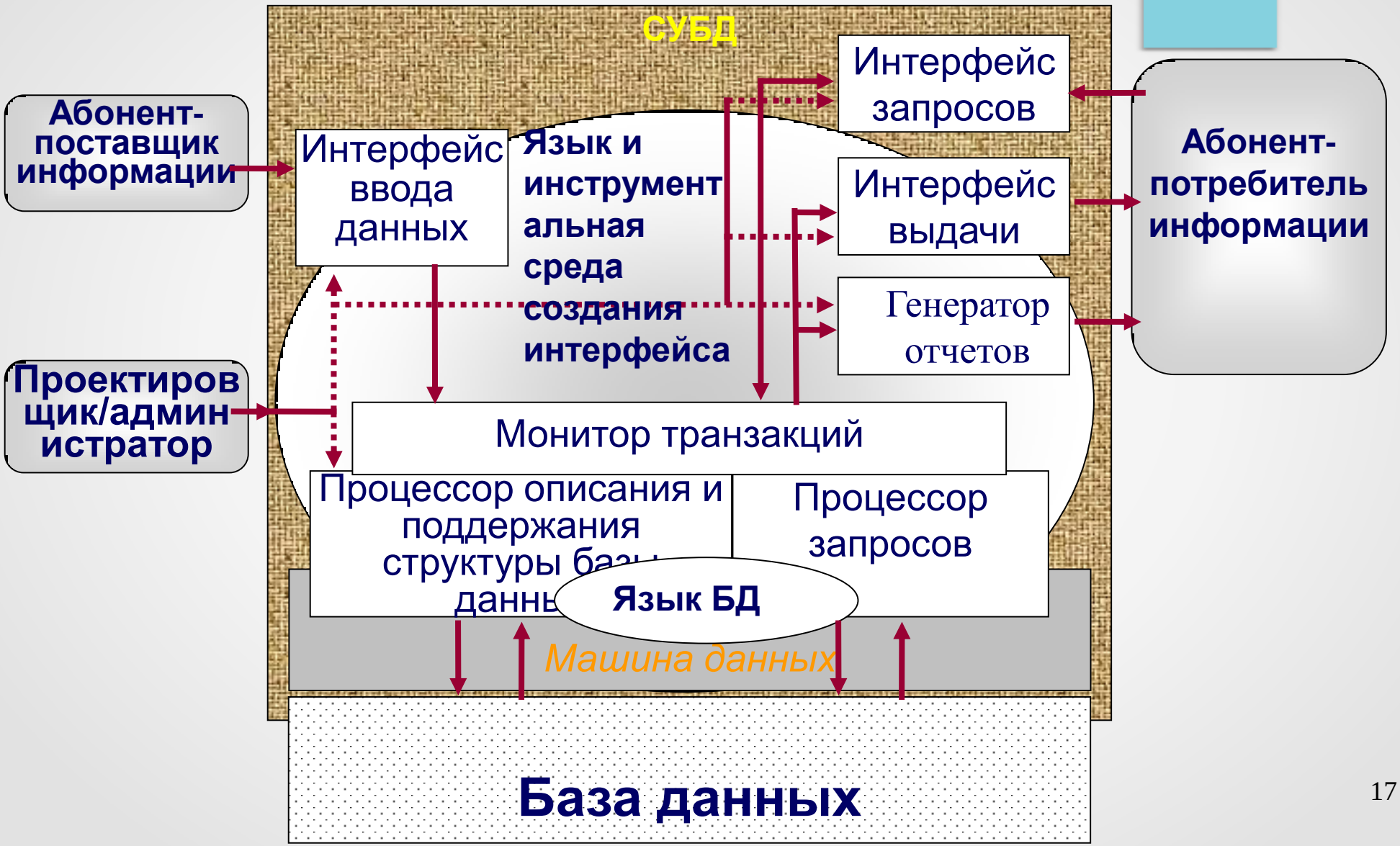
Сертификационный (удостоверяющий) центр

- доверенная третья сторона регистрирует абонентов открытых ключей
- изготавливает открытые и закрытые ключи и сертификаты открытых ключей
- обеспечивает доступ к сертификатам открытых ключей
- ведет справочник действующих и отозванных сертификатов

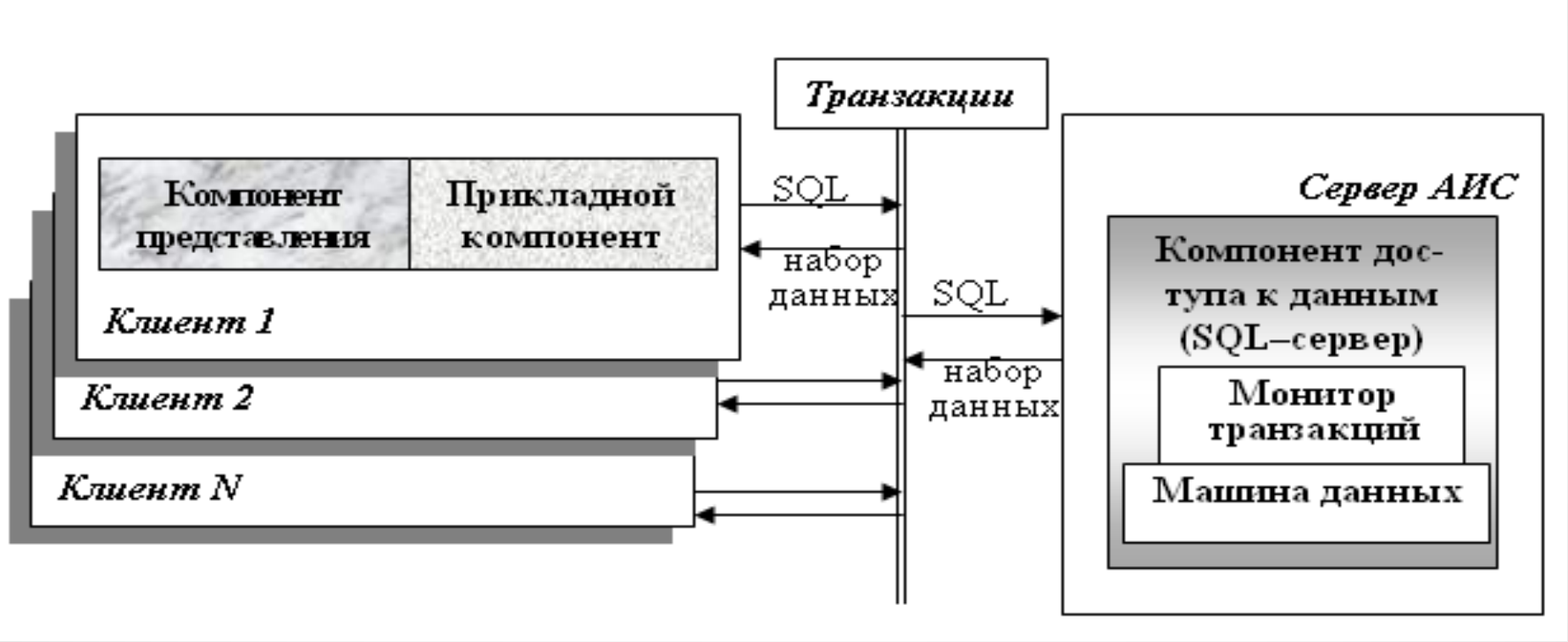
Иерархическая система сертификационных центров

- сертификат открытого ключа самого сертификационного центра выдает сертификационный центр более высшей иерархии

Структура СУБД



Структура СУБД



Транзакция - последовательная совокупность операций, имеющая отдельное смысловое значение по отношению к текущему состоянию базы данных

Совокупность функций СУБД по организации и управлению транзакциями - монитор транзакций

Структура СУБД. Виды нарушений целостности

Транзакции играют важную роль в механизме обеспечения СУБД ограничений целостности базы данных. Ограничения целостности непосредственно проверяются по завершению очередной транзакции. Если условия ограничений целостности данных не выполняются, то происходит "откат" транзакции (выполняется SQL–инструкция ROLLBACK), в противном случае транзакция фиксируется (выполняется SQL–инструкция COMMIT).

Виды нарушений целостности

Потерянные изменения - когда две транзакции одновременно изменяют один и тот же объект базы данных. В том случае, если в силу каких-либо причин, например, из–за нарушений целостности данных, происходит откат, скажем, второй транзакции, то вместе с этим отменяются и все изменения, внесенные в соответствующий период времени первой транзакцией. В результате первая еще не завершившаяся транзакция при повторном чтении объекта не "видит" своих ранее сделанных изменений данных.

Структура СУБД. Виды нарушений целостности

Виды нарушений целостности

"Грязные" данные - когда одна транзакция изменяет какой-либо объект данных, а другая транзакция в этот момент читает данные из того же объекта. Так как первая транзакция еще не завершена, и, следовательно, не проверена согласованность данных после проведенных, или вовсе еще только частично проведенных изменений, то вторая транзакция может "видеть" соответственно несогласованные, т.е. "грязные" данные.

Неповторяющиеся чтения - когда одна транзакция читает какой-либо объект базы данных, а другая до завершения первой его изменяет и успешно фиксируется. Если при этом первой, еще не завершенной, транзакции требуется повторно прочитать данный объект, то она "видит" его в другом состоянии, т.е. чтение не повторяется.

Механизмы изоляции транзакций

Два основных режима захватов:

совместный режим (*Shared*) - *захват по чтению*

монопольный режим (*eXclusive*) - *захват по записи*

Двухфазный протокол синхронизационных захватов (*блокировок*) *объектов* базы данных – 2PL (Two–Phase Locks)

(гранулирование" объектов захвата)

1-*я фаза - транзакция запрашивает и накапливает захваты необходимых объектов в соответствующем режиме,*

2-*я фаза – выполнение операций над захваченными объектами, фиксация изменений* (*или откат по соображениям целостности данных*), *освобождение захватов*

Возможность возникновения тупиковых ситуаций (*Deadlock*)

Автоматическое обнаружение (*распознавание*) *тупиковых ситуаций на построении и анализе графа ожидания транзакций*

Временные метки объектов базы данных

- Каждой транзакции приписывается временная метка, соответствующая моменту начала выполнения транзакции
- При выполнении операции над объектом транзакция "помечает"его своей меткой и типом операции (чтение или изменение)
- Если другой транзакции требуется операция над уже помеченным" объектом, то выполняются действия по следующему алгоритму:

  - проверяется, не закончилась ли транзакция, первой "пометившая" объект;
  - если первая транзакция закончилась, то вторая транзакция помечает его своей меткой и выполняет необходимые операции;
  - если первая транзакция не закончилась, то проверяется конфликтность операций (конфликтно любое сочетание, кроме "чтение–чтение");
  - если операции неконфликтны, то они выполняются для обеих транзакций, а объект до завершения операций помечается меткой более поздней, т.е. более молодой транзакции;
  - если операции конфликтны, то далее происходит откат более поздней транзакции и выполняется операция более ранней (старшей) транзакции, а после ее завершения, объект помечается меткой более молодой транзакции и цикл действий повторяется.

Доступность данных

- Отсутствие препятствий в доступе
- Сохранность данных
- Восстановление данных

  - Резервирование
  - Архивирование
  - Журналирование
  - Реплицирование

Резервирование, архивирование и журнализации данных

Резервирование

-организационно-технологическая система создания и обновления (поддержания актуальности) копий файлов БД
-установление и поддержания режима размещения, хранения и использования (доступа) копий БД для восстановления БД в случае сбоев и разрушений
-осуществляется либо средствами копирования файлов ОС, либо самой СУБД (специальными режимами работы СУБД или специальными утилитами СУБД)

«Горячее» резервирование

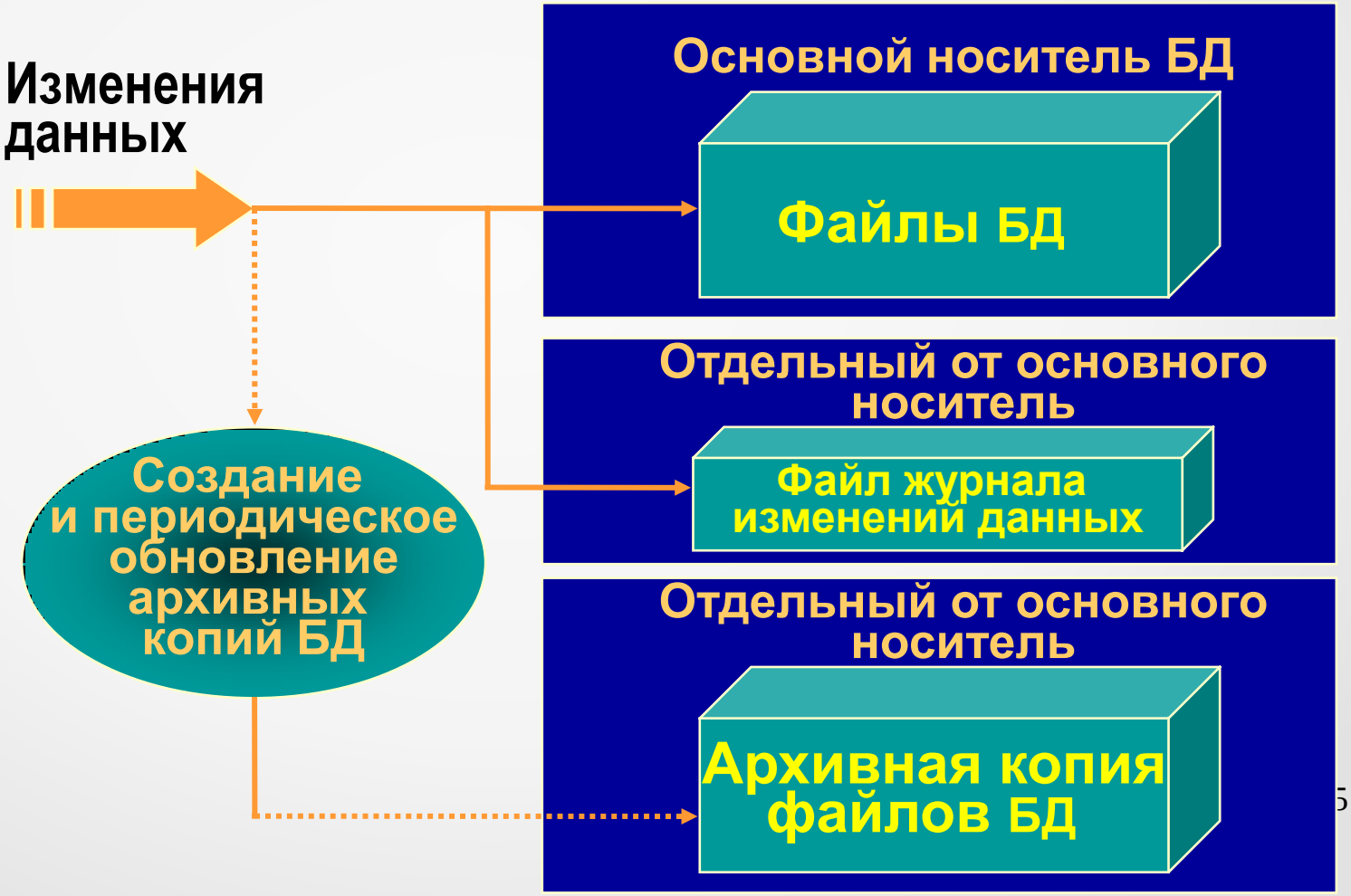
-постоянное и непрерывное функционирование 2-х или более равнозначных («зеркальных») копий БД в КС, относящихся к т.н. «системам реального времени»
-все изменения данных одновременно и параллельно фиксируются в зеркальных копиях БД
-при сбое одной копии функционирование КС обеспечивается другой «зеркальной» копией

Архивирование

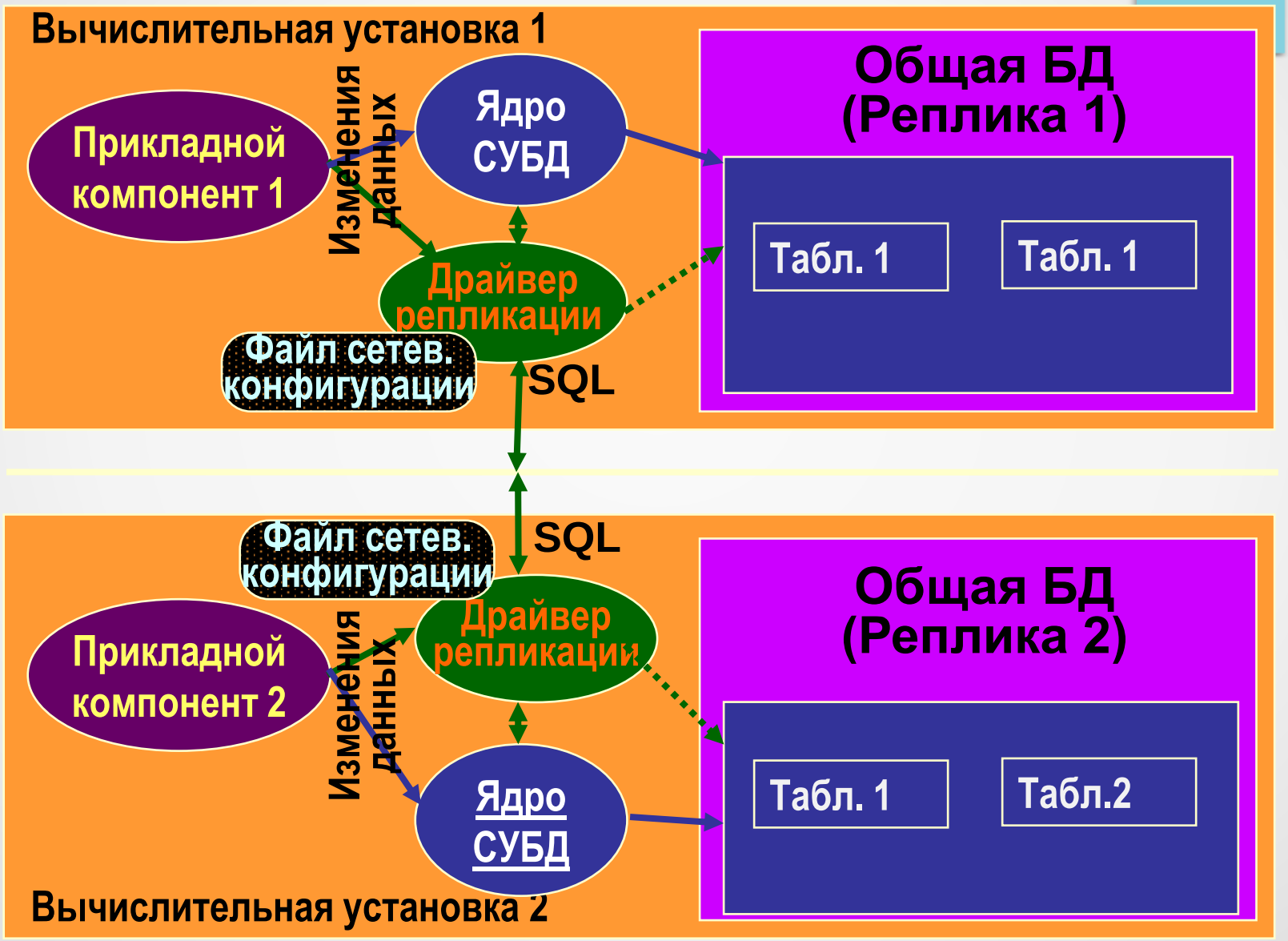
-по сути другое название системы резервирования данных, поскольку из-за большого размера файлов БД, создание резервных копий осуществляется с одновременным «сжатием» файлов данных
-сохраненная и «сжатая» копия БД называется «архивом» БД

Резервирование, архивирование и журнализация данных

Система ведения специальных журналов текущих изменений данных
а) для аудита действий пользователей КС,
б) для восстановления актуального состояния БД из существующего архива и произведенных с момента его создания изменений данных.



Программно-техническая структура систем репликации



Технологии и системы репликации данных

Обеспечение согласованного состояния данных.

-принцип непрерывного размножения обновлений (системы реального времени);

-принцип отложенных обновлений.

Обеспечения согласованного состояния структуры данных.

- Режим синхронной репликации изменений данных
- Режим асинхронной репликации изменений данных
- Системы с «главной» репликой
- Системы с частичными репликами

Технологии и системы репликации данных

Обеспечение непрерывности согласованного состояния данных

-системы синхронной репликации

-системы асинхронной репликации

Режим синхронной репликации изменений данных

-любая транзакция с любой рабочей станции сети осуществляется одновременно на всех репликах БД (фиксация транзакции производится только тогда, когда она успешно завершается одновременно на всех репликах системы)

-применяются аналогично клиент-серверным системам протоколы осуществления и фиксации транзакций

Проблемы и недостатки систем синхронной репликации

-снижение быстродействия обработки данных вследствие большого трафика данных в сети

-«тупики» при осуществлении транзакций (как в клиент-серверных системах)

Технологии и системы репликации данных

Обеспечение непрерывности согласованного состояния данных

Режим асинхронной репликации изменений данных

-транзакция на рабочих станциях осуществляются независимо друг от друга, в результате допускается текущая несогласованность состояния данных

-через определенные интервалы (по специальному графику, по специальным командам, в определенном, например во внерабочее, время и т.д.) осуществляется синхронизация реплик БД

-на рабочих станциях КС могут создаваться специальные хранилища данных репликации «накапливающие» изменения данных, поступающие с других рабочих станций

Проблемы и недостатки систем асинхронной репликации

-не могут применяться в КС, с высокой динамикой изменения данных

-в результате синхронизации реплик могут наблюдаться «потерянные изменения» при взаимном затирании изменений одних и тех объектов на разных репликах

Технологии и системы репликации данных

Обеспечение непрерывности согласованного состояния структуры данных

-системы с «главной» репликой

-системы с частичными репликами

Системы с «главной» репликой

-одна из реплик объявляется «главной» и только на ней допускается изменение структуры данных (добавление/удаление таблиц, изменение схемы таблиц)

-по принципу асинхронной репликации осуществляется тиражирование соответствующих изменений структуры данных по всем репликам системы

Системы с частичными репликами

-в каждой локальной БД определяются перечень реплицируемых объектов, в результате локальные БД «одинаковы» только в определенной части

-синхронизация реплик может осуществляется в синхронном и асинхронном (отложенном) режиме

-могут создаваться распределенные КС с функционально обоснованной схемой ввода и тиражирования данных, например данные в таблицу «Документы» вводятся в реплике секретариата и тиражируются по всем репликам, находящихся в др. подразделениях, данные в таблицу «Сотрудники» - в реплике кадрового подразделения и т.д.