Институт системного программирования
Российской академии наук

Технология виртуализации: акцент на защиту приложений от несанкционированных действий операционной системы

Яковенко Павел
yak@ispras.ru

ISP RAS

- «Массовые операционные системы характеризуются двумя чертами, которые делают их малонадежными и слабо защищенными: они слишком объемные и имеют крайне плохую изоляцию последствий отказов в отдельных компонентах»

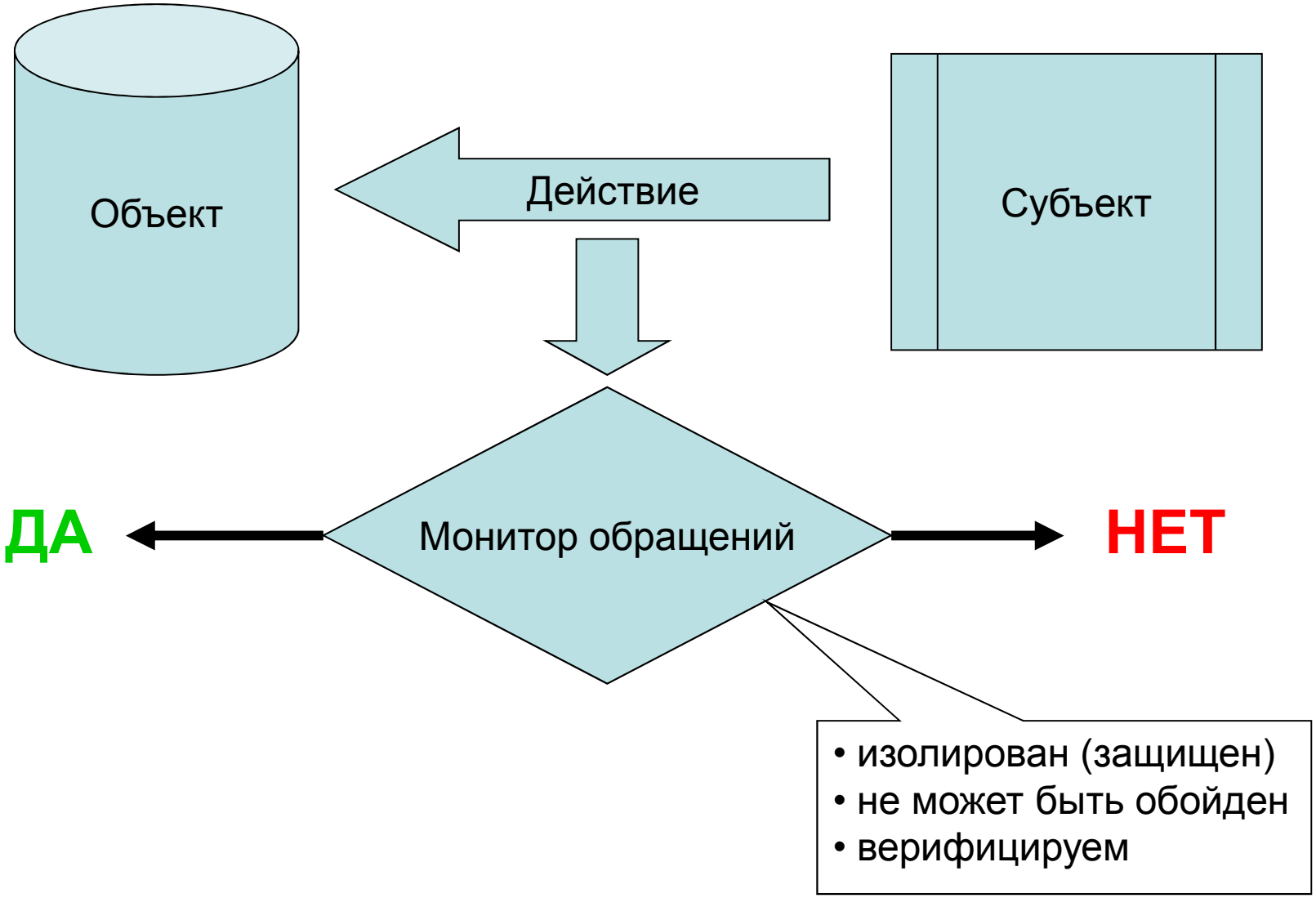
  Таненбаум и др. (2006)

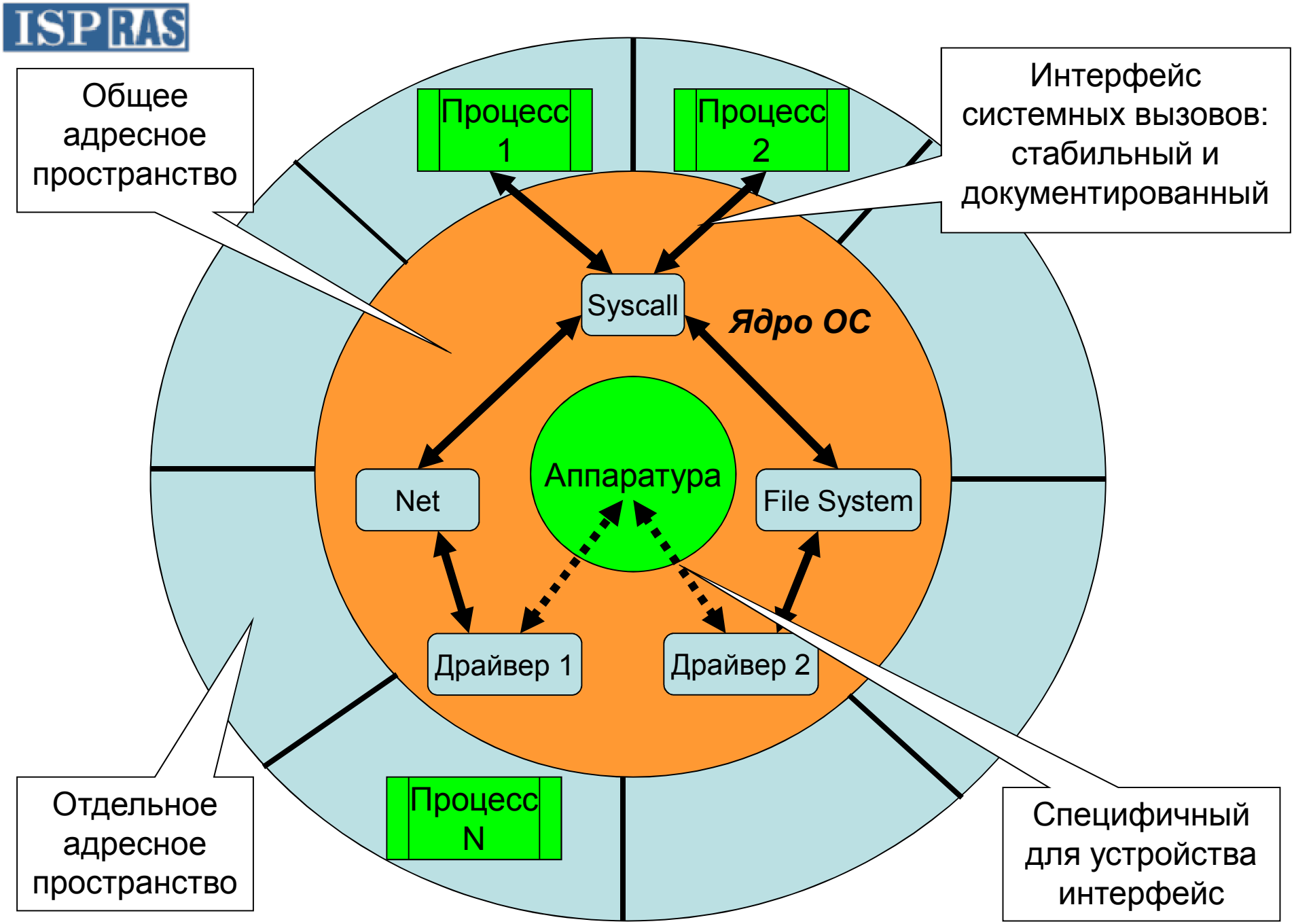
- «С течением времени цифровые системы будут только усложняться, а сложность – худший враг безопасности»

  Шнайер (2000)

- «Всего лишь одна логическая ошибка в коде операционной системы может полностью свести на нет работу всех защитных механизмов»

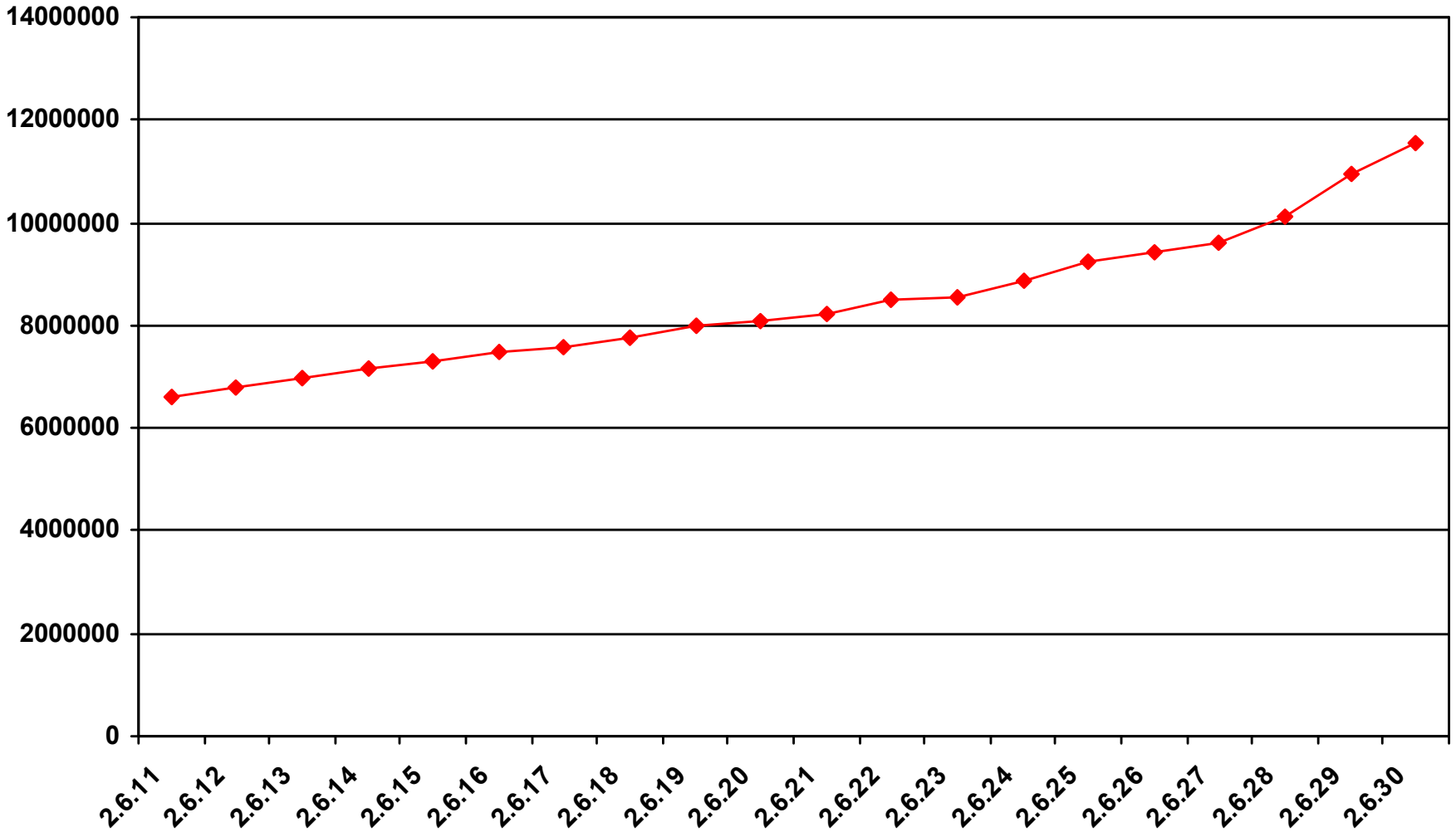
  Мэдник и *Донован* (1973)


Объект
Действие
Субъект
ДА
Монитор обращений
НЕТ
• изолирован (защищен)
• не может быть обойден
• верифицируем


ISP RAS
Общее адресное пространство
Процесс 1
Процесс 2
Интерфейс системных вызовов: стабильный и документированный
Syscall
Ядро ОС
Аппаратура
Net
File System
Драйвер 1
Драйвер 2
Процесс N
Отдельное адресное пространство
Специфичный для устройства интерфейс

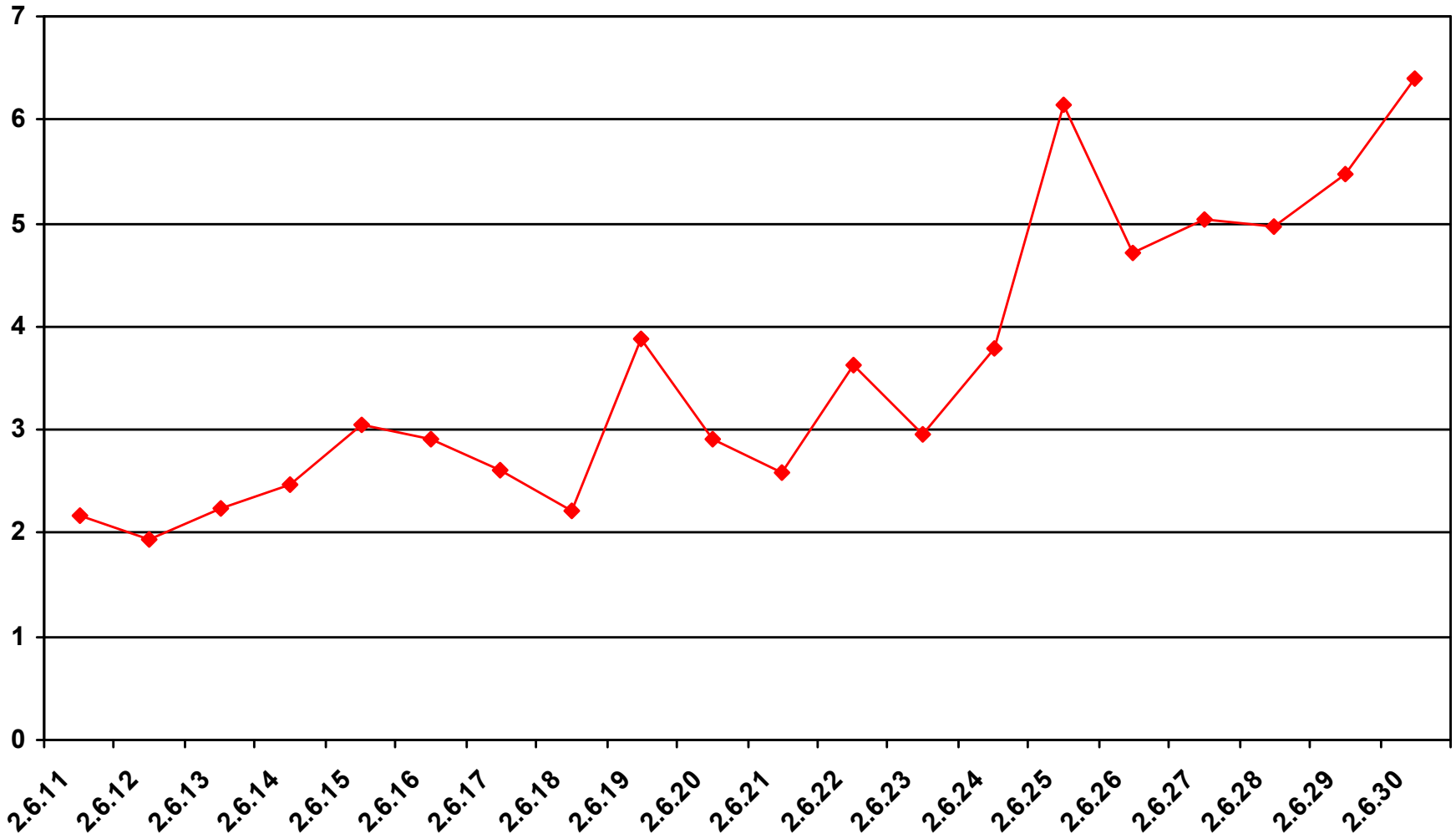
ISP RAS

Linux: кол-во строк кода



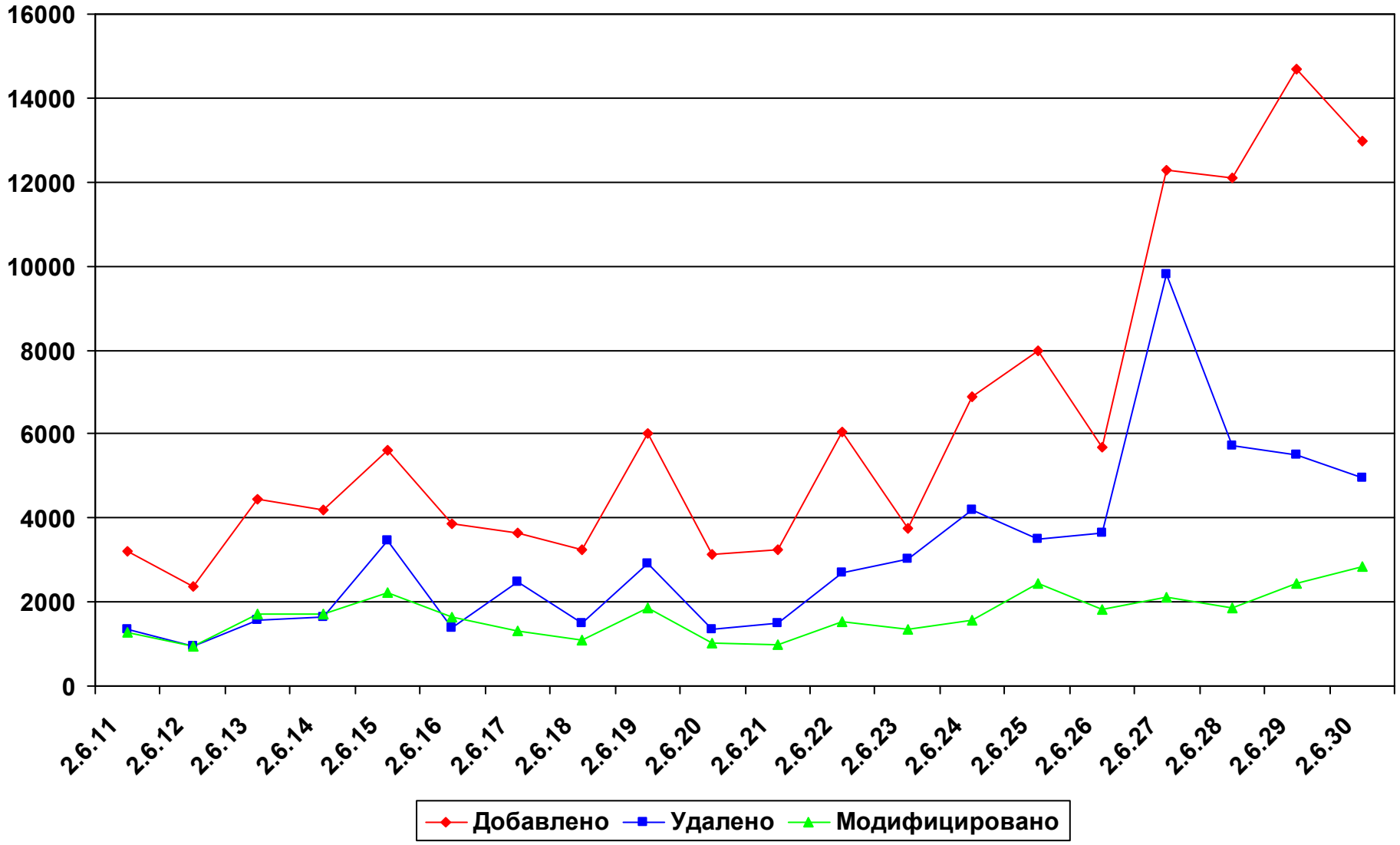
ISP RAS

Linux: изменений (транзакций) в час



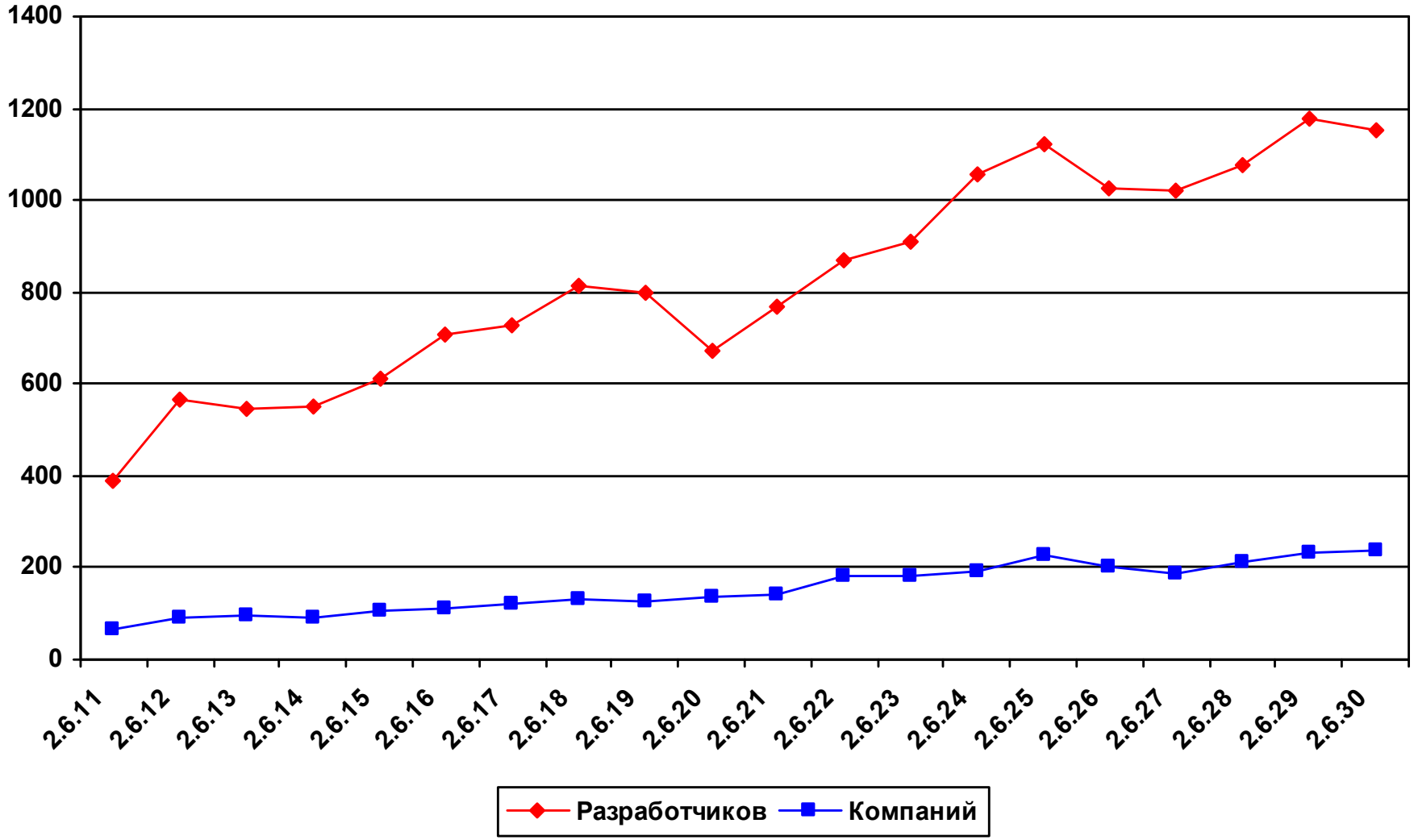
ISP RAS

Linux: изменений (строк кода) в день



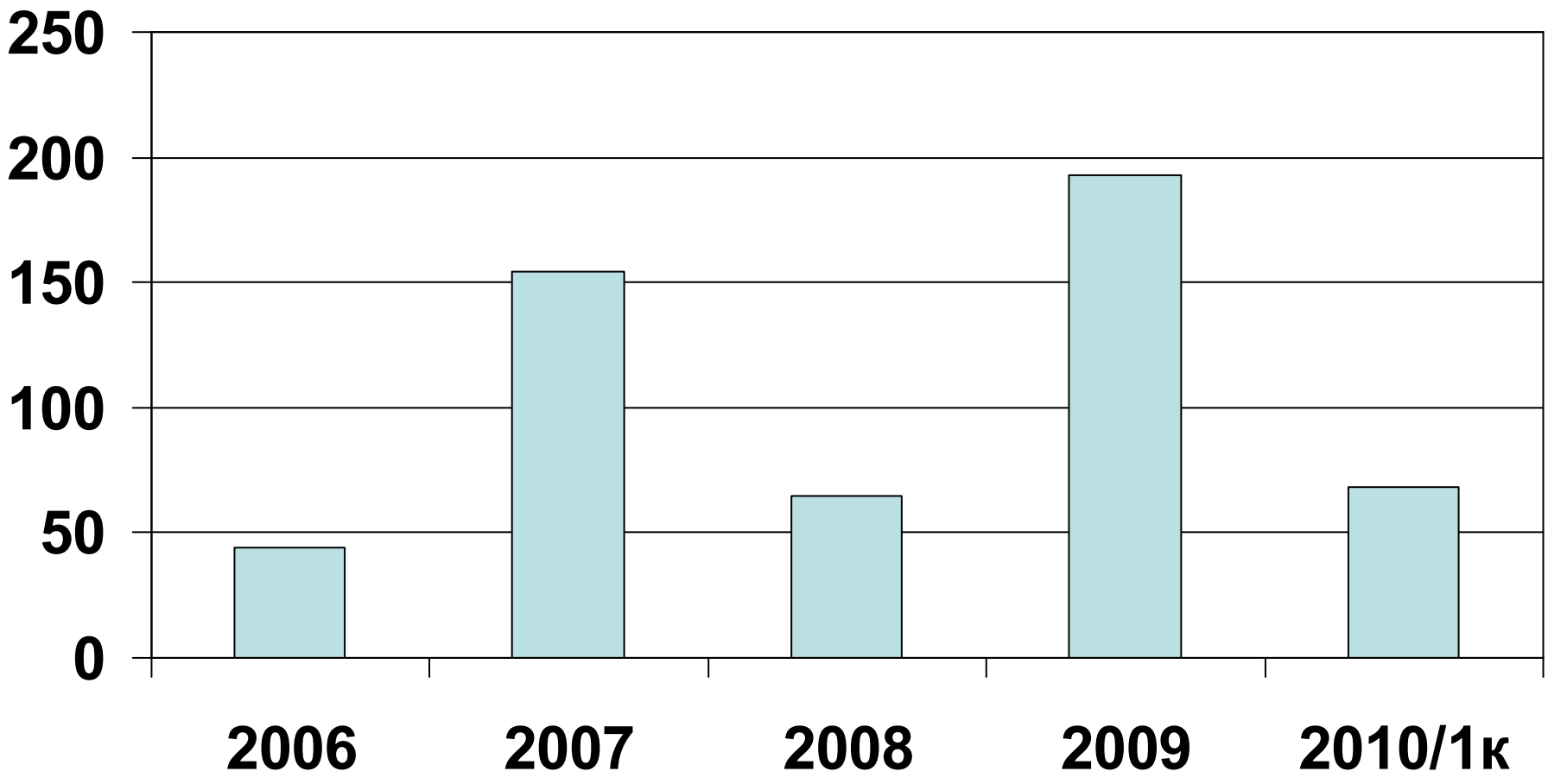
ISP RAS

Linux: число разработчиков



ISP RAS

Linux: уязвимости ядра (по данным SecurityFocus)

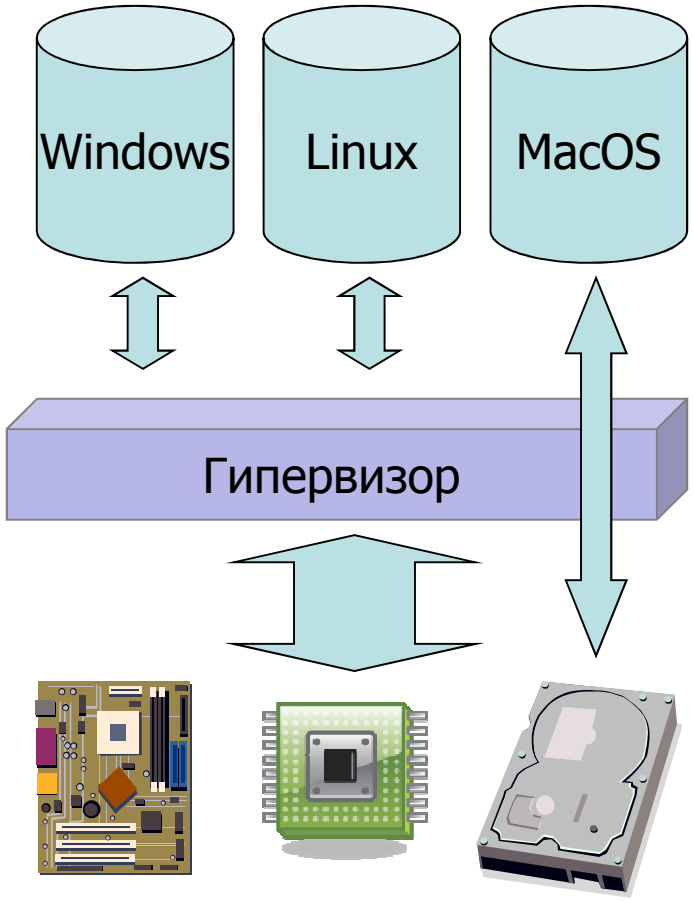


ISP RAS

- Ядро в массовых ОС построено на базе монолитной архитектуры
  - большой объем кода выполняется в привилегированном режиме (в т.ч. драйверы периферийных устройств)
  - большой объем кода + монолитность ядра → недостаточная надежность ОС
- Недостаточная надежность → слабая защищенность
  - уязвимости в ядре
  - недокументированные возможности
- Современные массовые ОС предоставляют удобную среду для пользователя и разработчика
  - развитые пользовательские приложения
  - большое количество библиотек для разработчика
  - широкая поддержка периферийных устройств

Задача

Надежный способ контроля доступа к ресурсам в условиях недоверенной операционной системы без модификации программного обеспечения



- Бинарная трансляция
  - VMware
- Паравиртуализация
  - XEN
- Аппаратные расширения
  - AMD-SVM, Intel-VT

Паравиртуализация

```
void switch_mm(struct mm_struct *prev, struct mm_struct *next, struct task_struct *tsk)
{
  [...]

  /* Re-load page tables */
  load_cr3(next->pgd);

  /*
   * load the LDT, if the LDT is different:
   */
  if (unlikely(prev->context.ldt != next->context.ldt))
    load_LDT_nolock(&next->context, cpu);
}
```

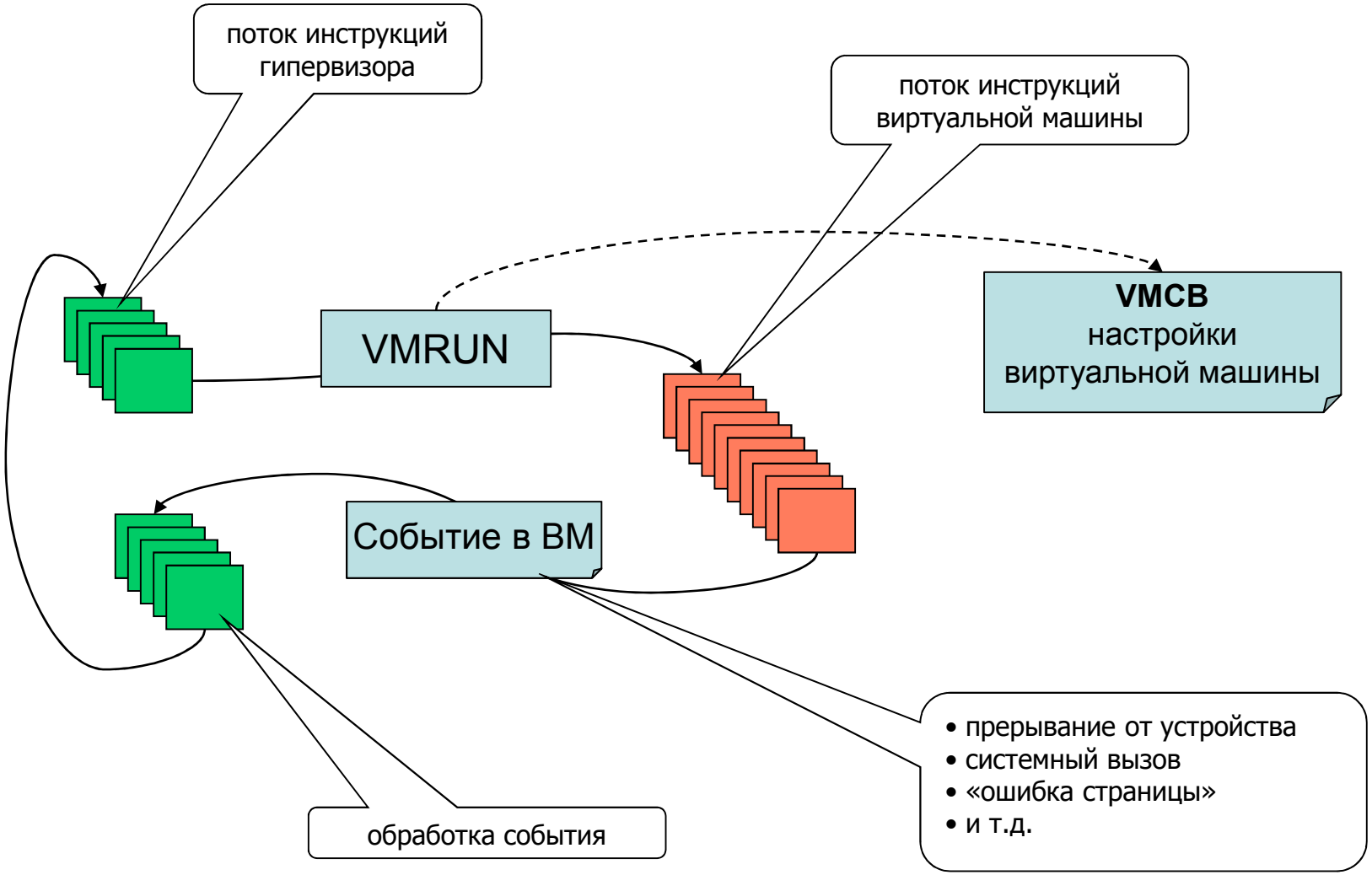
Прямой доступ к системным регистрам

```
void switch_mm(struct mm_struct *prev, struct mm_struct *next, struct task_struct *tsk)
{
  [...]

  op->cmd = MMUEXT_NEW_BASEPTR;
  op->arg1.mfn = pfn_to_mfn(__pa(next->pgd) >> PAGE_SHIFT);
  op++;

  if (unlikely(prev->context.ldt != next->context.ldt))
  {
    op->cmd = MMUEXT_SET_LDT;
    op->arg1.linear_addr = (unsigned long)next->context.ldt;
    op->arg2.nr_ents     = next->context.size;
    op++;
  }

  HYPERVISOR_mmuext_op(_op, op-_op, NULL, DOMID_SELF);
}
```

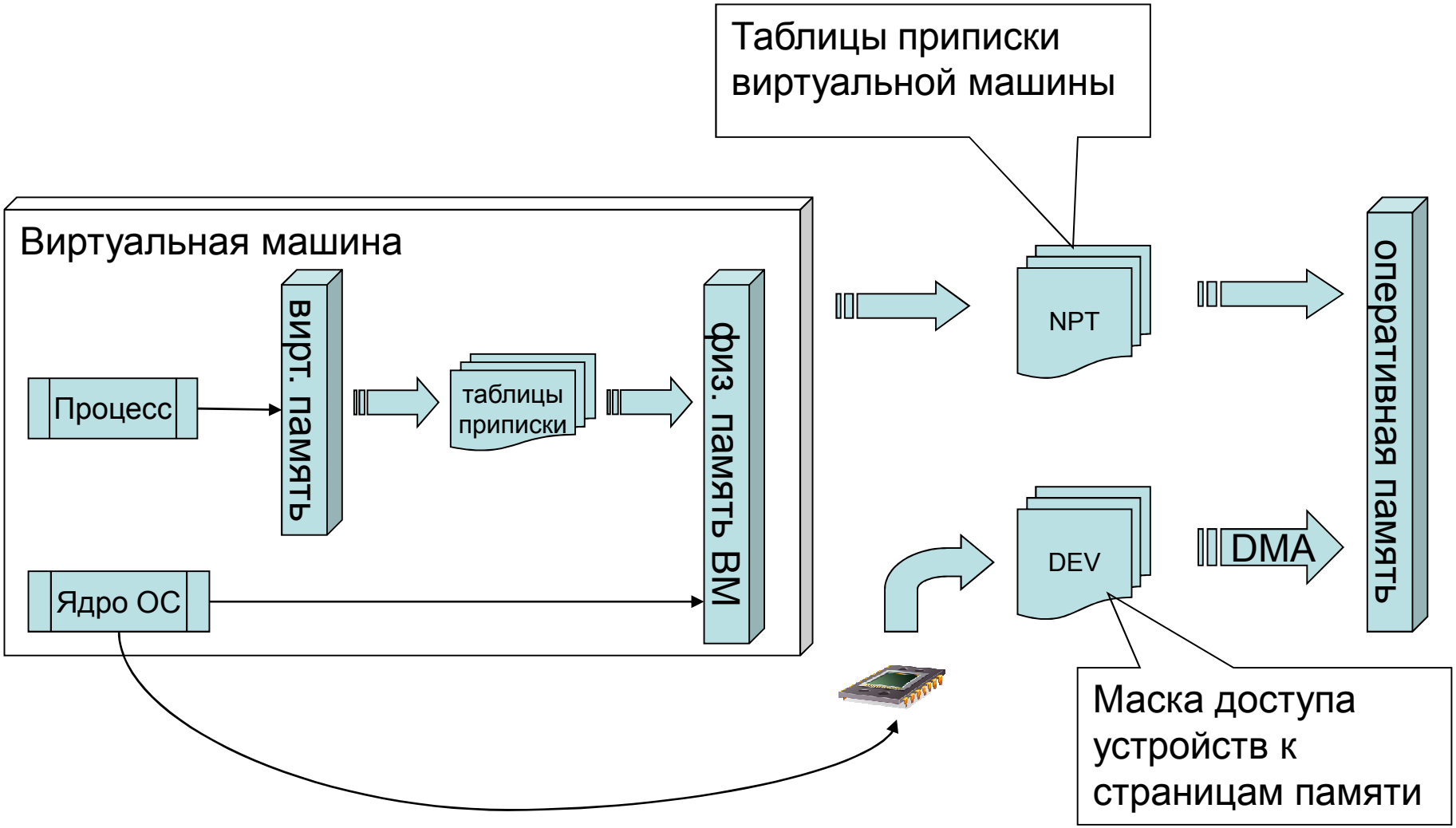
Идентификатор гипервызова

Выполнение гипервызова

Расширения в аппаратуре-1 (AMD)



Расширения в аппаратуре-2 (AMD)



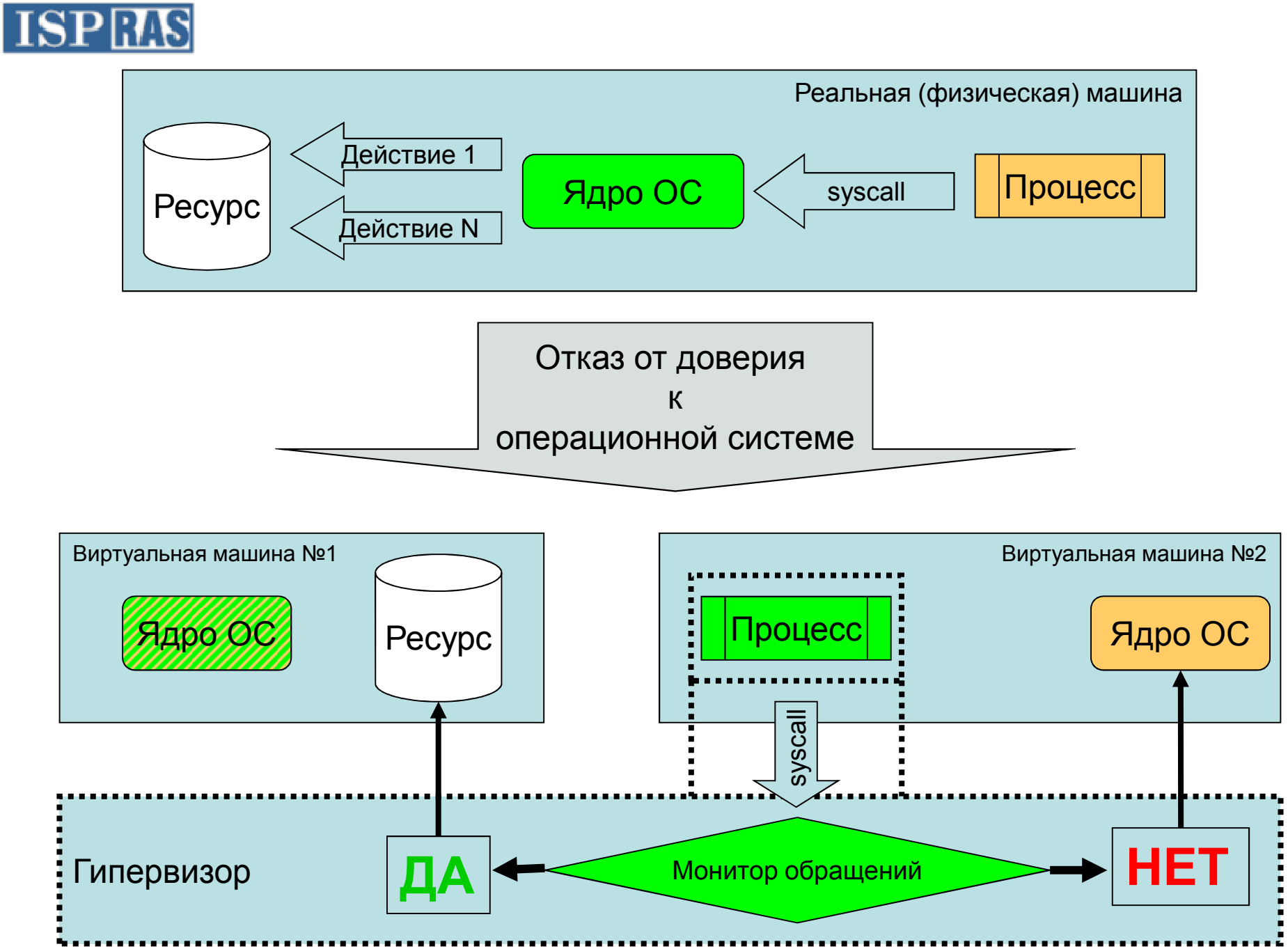

Реальная (физическая) машина
Ресурс
Действие 1
Действие N
Ядро ОС
syscall
Процесс
Отказ от доверия
к
операционной системе
Виртуальная машина №1
Ядро ОС
Ресурс
Виртуальная машина №2
Процесс
Ядро ОС
syscall
Гипервизор
ДА
Монитор обращений
НЕТ

Схема 1: «Коммутатор» (конфиденциальность)

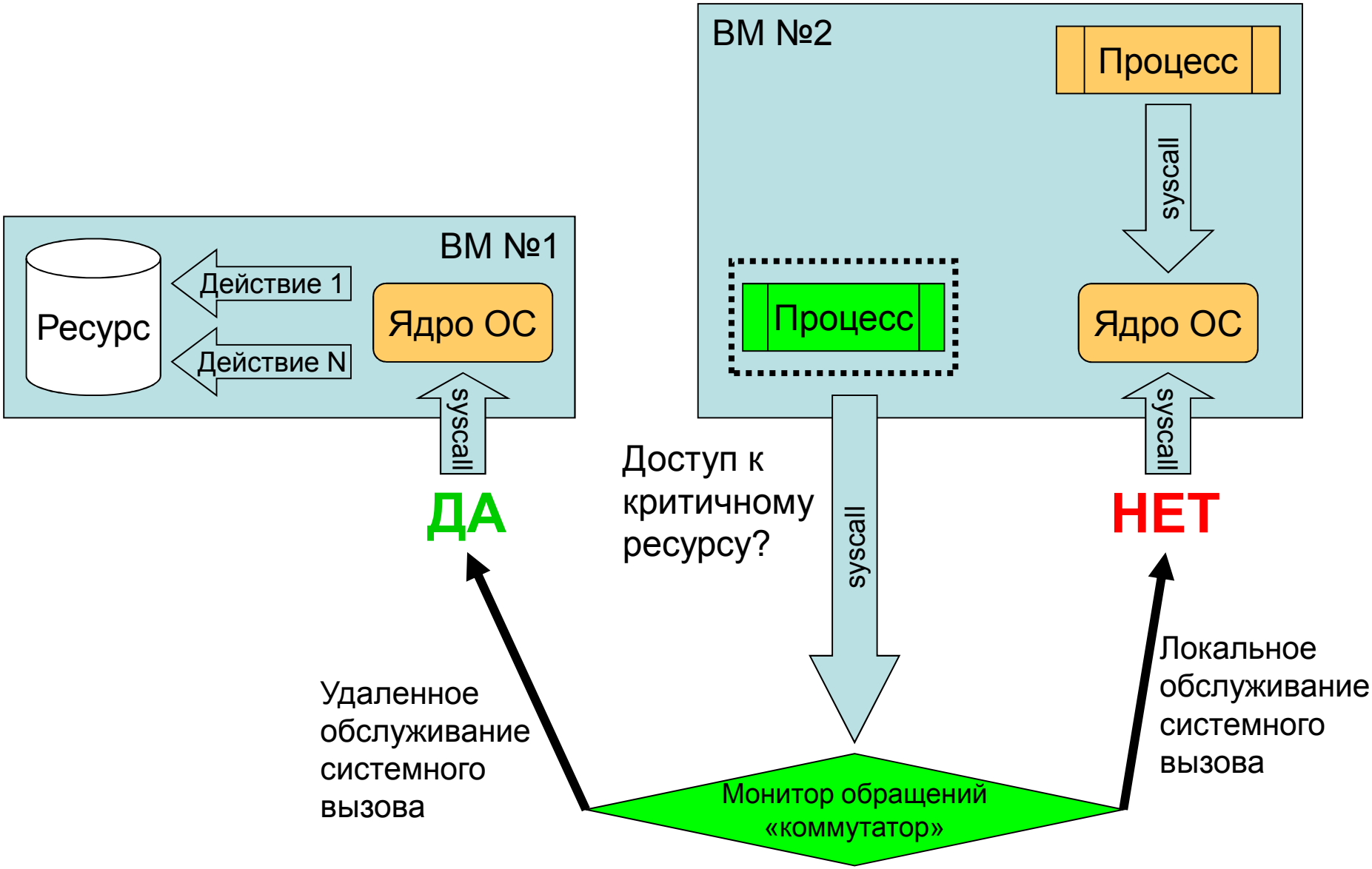
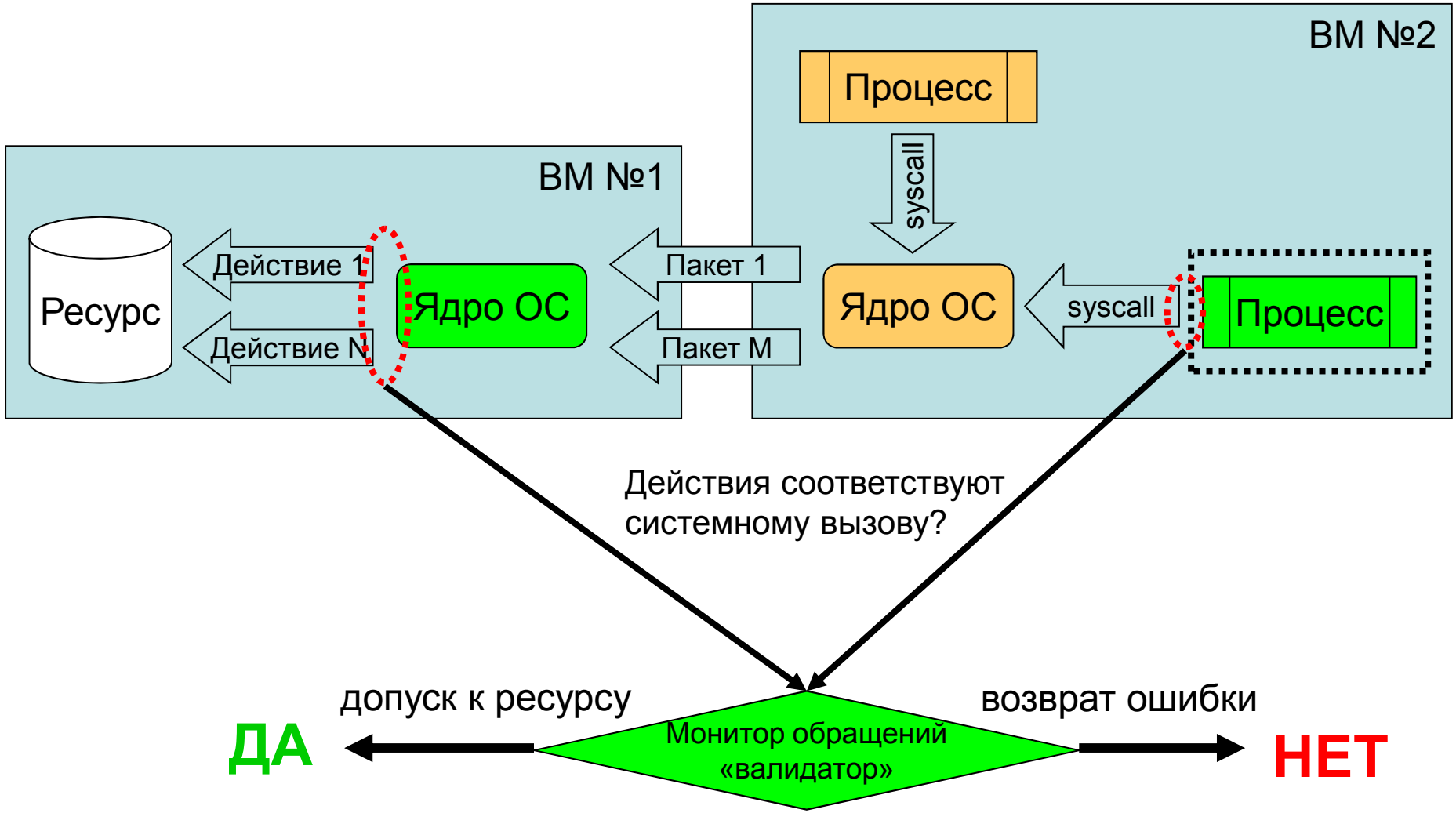
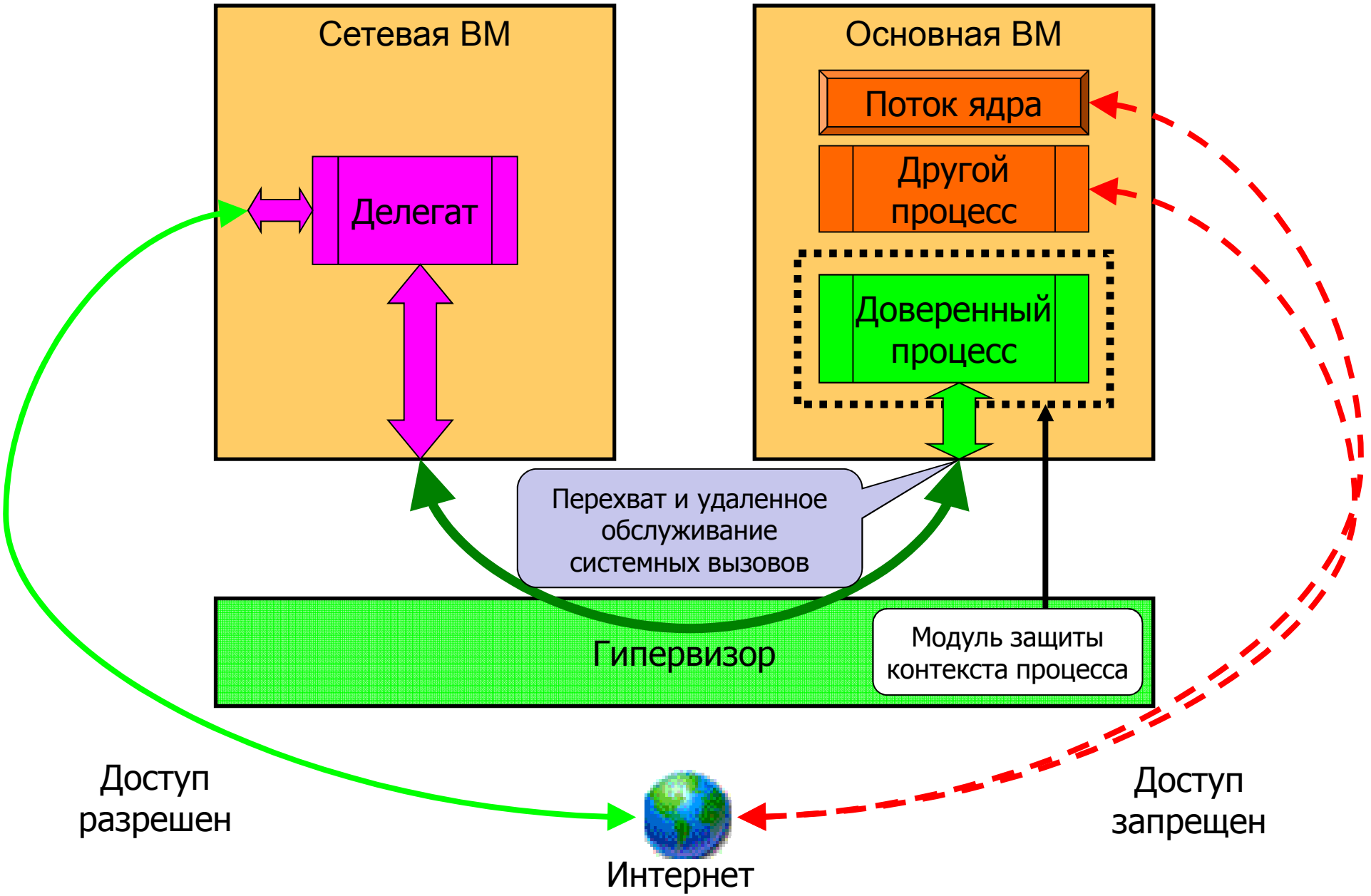


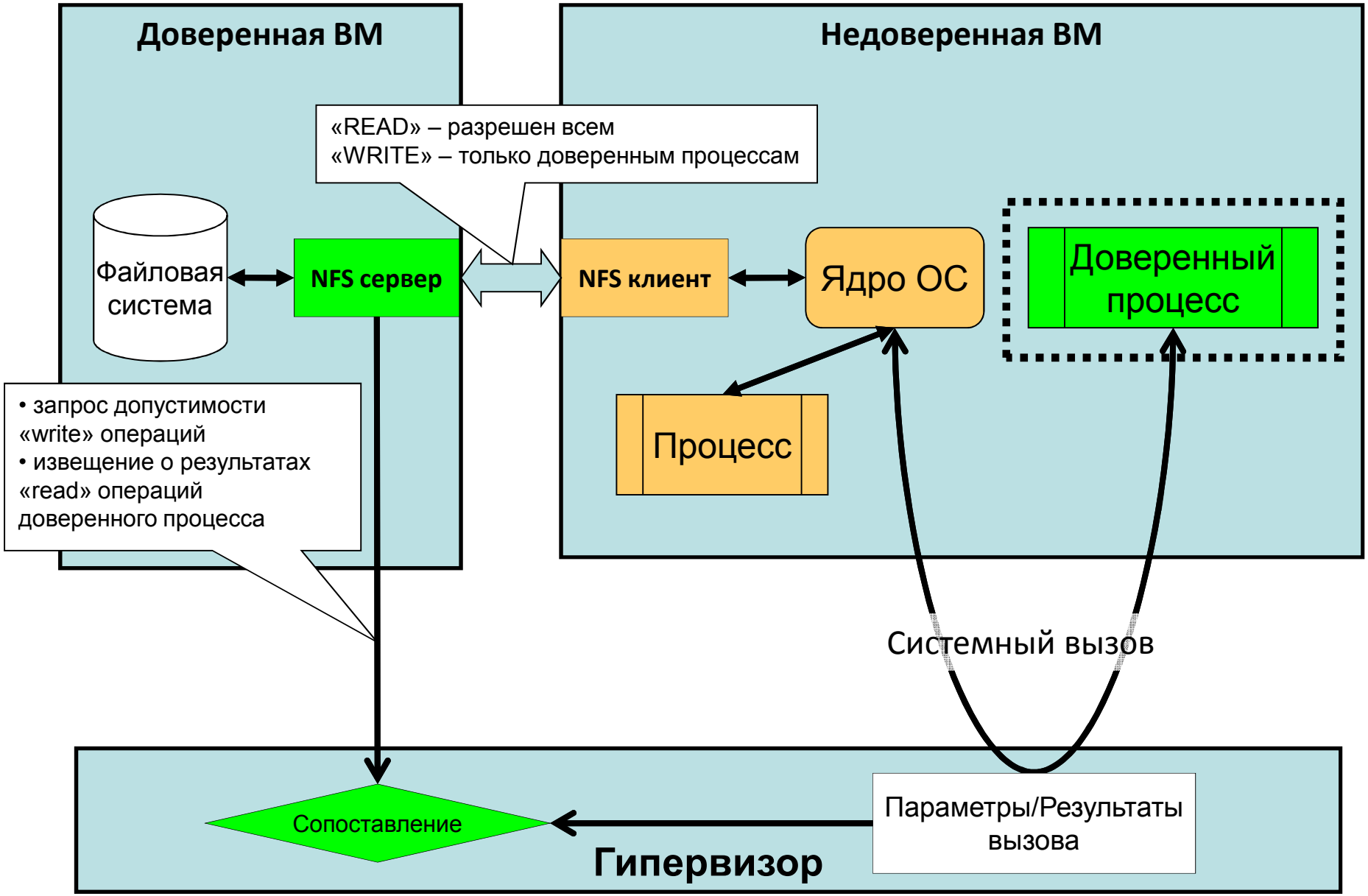
Схема 2: «Валидатор» (целостность)



Пример: защита конфиденциальности файлов



Пример: защита целостности файлов



ISP RAS

- Гипервизор предоставляет доступ к ресурсам другой ВМ (критичным ресурсам) только *авторизованным* доверенным приложениям
  - Авторизация производится путем предоставления хэш-кодов для исполняемых файлов приложения
- Доверенные процессы выполняются под управлением скомпрометированной операционной системы
  - $\Rightarrow$ обеспечение доступности не представляется возможным
- Принцип Керкгоффса «Система не должна требовать секретности, на случай, если она попадёт в руки врага»
  - Предполагается, что схема работы системы защиты известна злоумышленнику
- Доверенные процессы должны быть ***защищены*** от несанкционированного использования их контекста для доступа к критичному ресурсу
  - Предоставляемая защита должна быть независима от встроенных механизмов защиты ОС
  - Более того, необходимо защита должна предусматривать возможность вредоносного воздействия со стороны кода в ядре ОС

ISP RAS

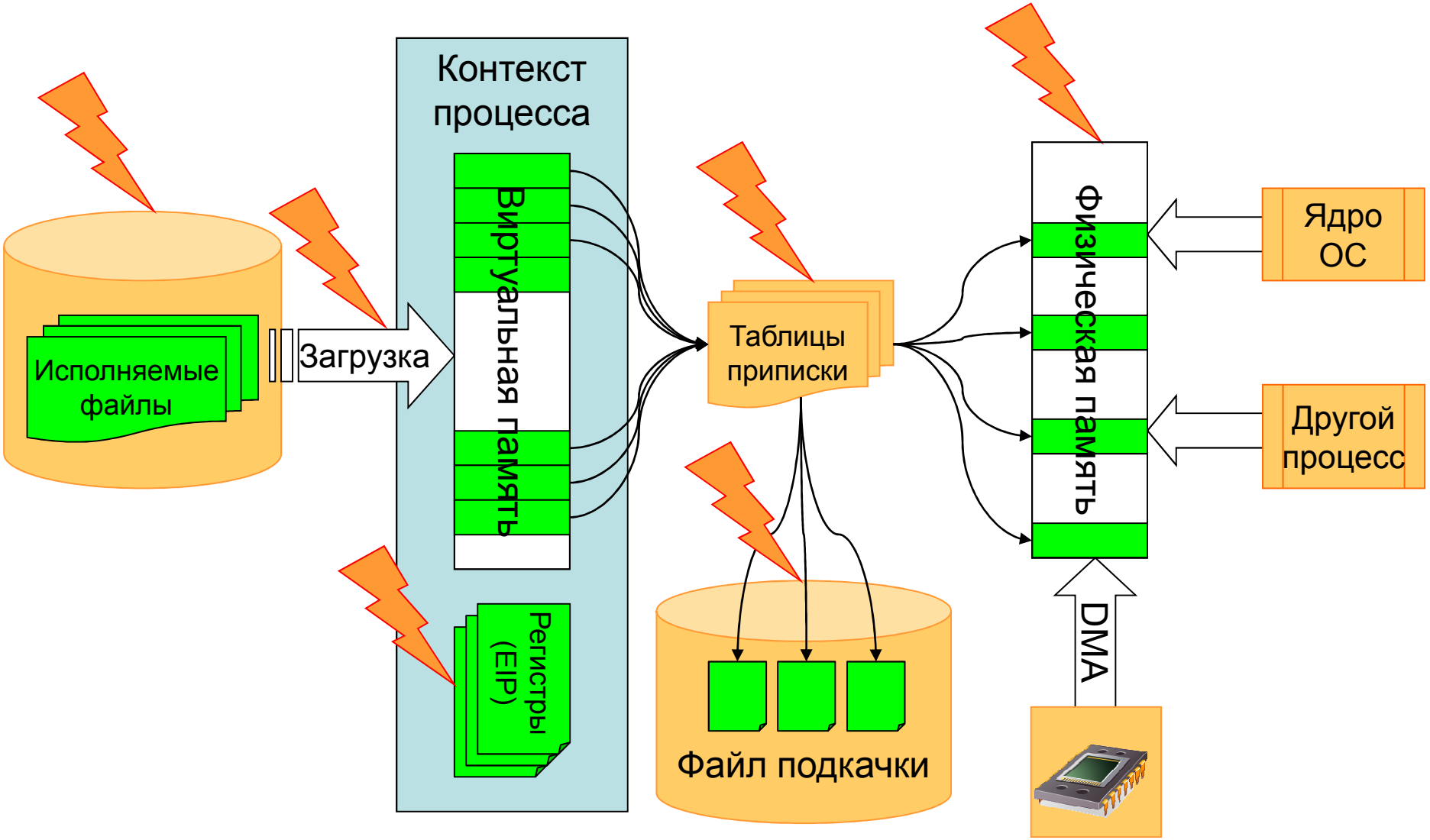
- Характер защиты
  - обнаружение несанкционированной модификации контекста процесса *до того* как измененный код получит управление
  - непрерывно на все протяжении от момента запуска до завершения процесса

- Защищаемые элементы контекста
  - содержимое регистров процессора
  - адресное пространство процесса (код, данные, стек)
    - в т.ч. таблицы связывания объектных модулей (PLT, GOT)
  - контролируются содержимое ***всех*** виртуальных страниц процесса, начиная с первого обращения процесса к ним

- Требования целостности
  - стартовое состояние процесса (в точке запуска) должно соответствовать ожидаемому (эталонному)
  - дальнейшая модификация контекста чужим кодом (в частности, ядром ОС) допускается только в явно указанных ***самим процессом*** областях памяти
  - данные области специфицируются системными вызовами (например, *read*)

- Реакция при обнаружении нарушения целостности
  - лишение статуса «доверенного» процесса
  - отказ в доступе к ресурсам другой ВМ
  - восстановление контекста к целостному виду не требуется, но технически возможно

Возможные точки модификации контекста



ISP RAS

Доверенный режим выполнения процесса

- Процесс считается доверенным, пока состояние его контекста удовлетворяет условиям целостности **C = (R,M)**
  - R = {(<регистр>, <значение>)}
  - M = {(<виртуальная страница>, <SHA-1>)}
  - виртуальная страница – блок памяти размером 4Кб

- Доступ к критичному ресурсу допускается только посредством системного вызова:
  1. в контексте *доверенного* процесса
  2. при активированном гипервизором *доверенном* режиме
  3. из *пользовательского* режима

- Вход в доверенный режим при передачи управления «ОС→Процесс» при выполнении условий **C**
  - фактически контролируется операционной системой!
  - проверка целостности страницы памяти V откладывается до фактического доступа к ней

- Выход из доверенного режима при передаче управления ОС
  - надежно контролируется гипервизором

- ОС может передать управление доверенному процессу нестандартным образом, но тогда гипервизор не включит для процесса доверенный режим
  - процесс будет рассматриваться как «недоверенный»
  - системный вызов по доступу к ресурсу будет проигнорирован
  - в ходе выполнения процесса произойдет естественное рассогласование фактического его состояния и условий C, что будет обнаружено при последующей проверке

Поток управления

ISP RAS

- Стартовый набор ограничений задается «регистрационной» информацией программы
- Регистрационная информация содержит контрольные суммы (SHA-1) для всех объектных модулей приложения
  - в т.ч. ld.so для динамически компонуемых приложений
- Регистрационная информация не привязана к конкретному имени или расположению приложения в файловой системе
  - у каждого приложения есть уникальный ID
- Запуск доверенного приложения осуществляется специальной программой «монитором», который сообщает гипервизору ID приложения
  - монитор принимает в качестве параметра «паспорт» задачи

```
struct mem_page_desc ssh_pages[459] =
{
  {0x8048000, {0x6f,0x50,0x89,0x16,0x29,0xa3,0xf1,0x2e,0xb0,...},
  ...
  {0x8212000, {0x1c,0xea,0xf7,0x3d,0xf4,0x0e,0x53,0x1d,0xf3,...}
};
struct exec_object_desc trusted_ssh =
{
  .entryPoint = 0x8048180,
  .pages = ssh_pages,
  .nPages = 459
};
struct trusted_object_desc trusted_objects[]
{
  {444, &trusted_ssh}
};
```

```
[STARTER]
/usr/local/bin/scp
[ARGUMENTS]
myfile user@host:myfile
[TRUSTED_OBJECTS]
444:/usr/local/bin/ssh
444:ssh
[CANCEL_HANDLER]
444:0x8049af3
```

```
static struct trusted_object_desc *trusted_ssh_objects[24] =
{
  &trusted_ssh_linux_gate_so_1,
  &trusted_ssh_libresolv_so_2,
  &trusted_ssh_libcrypto_so_7,
  &trusted_ssh_libutil_so_1,
  &trusted_ssh_libz_so_1,
  &trusted_ssh_libnsl_so_1,
  &trusted_ssh_libcrypt_so_1,
  &trusted_ssh_libnss3_so,
  &trusted_ssh_libgssapi_krb5_so_2,
  &trusted_ssh_libkrb5_so_3,
  &trusted_ssh_libk5crypto_so_3,
  &trusted_ssh_libcom_err_so_2,
  &trusted_ssh_libc_so_6,
  &trusted_ssh_libplc4_so,
  &trusted_ssh_libdl_so_2,
  &trusted_ssh_libnssutil3_so,
  &trusted_ssh_libplds4_so,
  &trusted_ssh_libnspr4_so,
  &trusted_ssh_libpthread_so_0,
  &trusted_ssh_libkrb5support_so_0,
  &trusted_ssh_libkeyutils_so_1,
  &trusted_ssh_ld_2_8_so,
  &trusted_ssh_libselinux_so_1,
  &trusted_ssh
};
```

```
static struct trusted_object_desc trusted_ssh_ld_2_8_so =
{
  .name = "ld-2.8.so",
  .loadAddress = 0x2b6000,
  .entryPoint = 0x2b6830,
  .cancelAddr = 0x2b6812,
  .isDynamic = 1,
  .pages = trusted_ssh_ld_2_8_so_pages,
  .nPages = 30
};
```

```
static struct trusted_vpage_desc trusted_ssh_ld_2_8_so_pages[30] =
{
  {0x2b6000, {0x6a,...,0x67}},
  {0x2b7000, {0x1f,...,0x38}},
  {0x2b8000, {0xb0,...,0x09}},
  ...
  {0x2d2000, {0xb9,...,0x14}},
  {0x2d3000, {0x2d,...,0xba}}
};
```

20 байт

```
static struct trusted_program_desc trusted_ssh_program =
{
  .entry = &trusted_ssh_ld_2_8_so,
  .objects = trusted_ssh_objects,
  .nObjects = 24
};
```

ISP RAS

- Домен защиты *физической* памяти **D = {(P,A)}**
  - P – адрес физической страницы, A – ограничение доступа к ней
  - A = <нет доступа> | <только чтение> | <полный доступ>
  - Реализуется на базе расширенных страниц приписки (NPT) и маски доступа устройств (DEV)

- Домены:
  - *ограничивают* доступ по записи к страницам оперативной памяти занятыми информацией доверенных процессов
  - обеспечивают *своевременную* проверку условий С доверенных процессов
  - отдельный домен для каждого доверенного процесса
  - общий домен для всех недоверенных процессов и ядра ОС

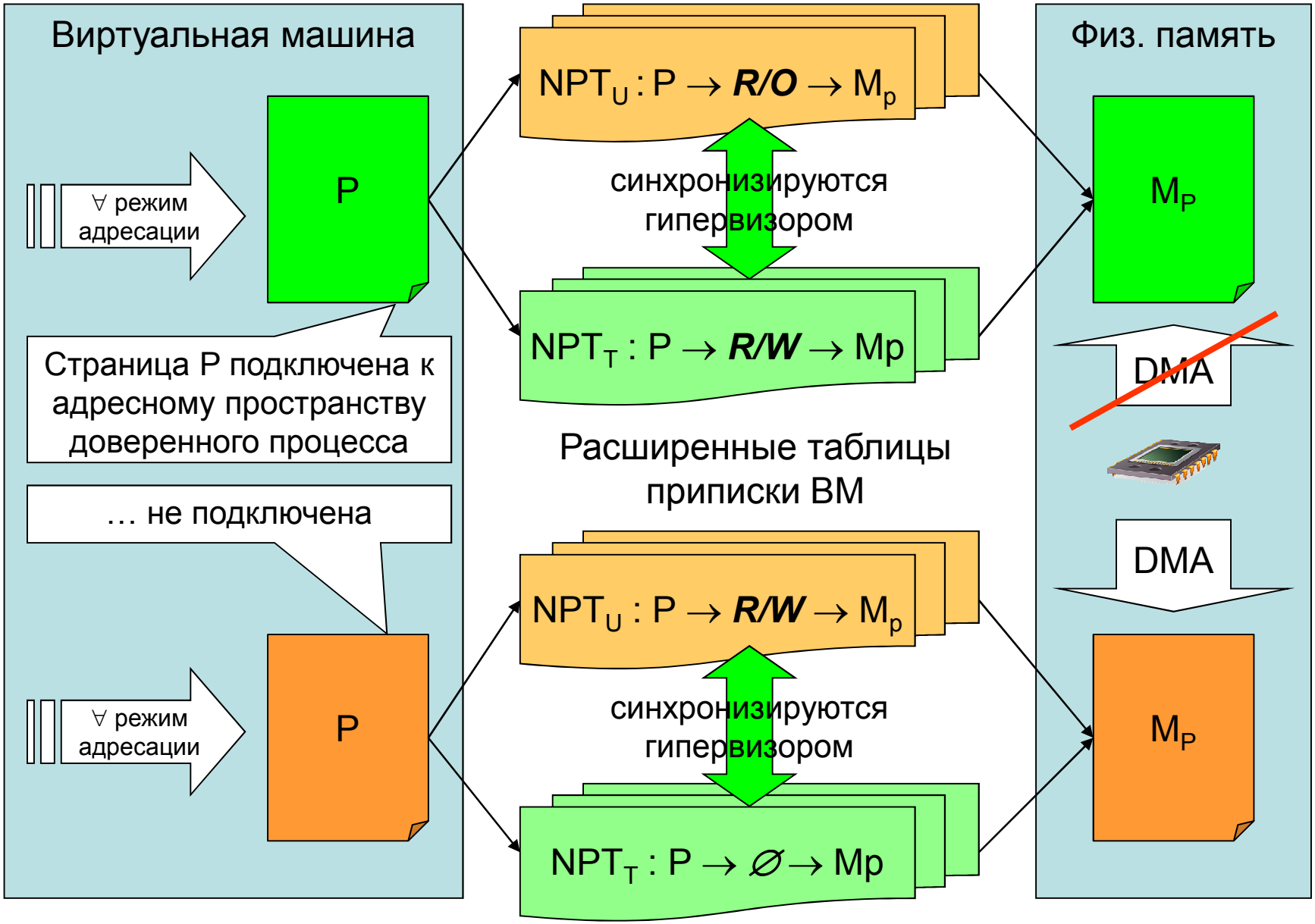
- В каждый момент времени активен только один из доменов
  - соответственно выполняемому на процессоре коду

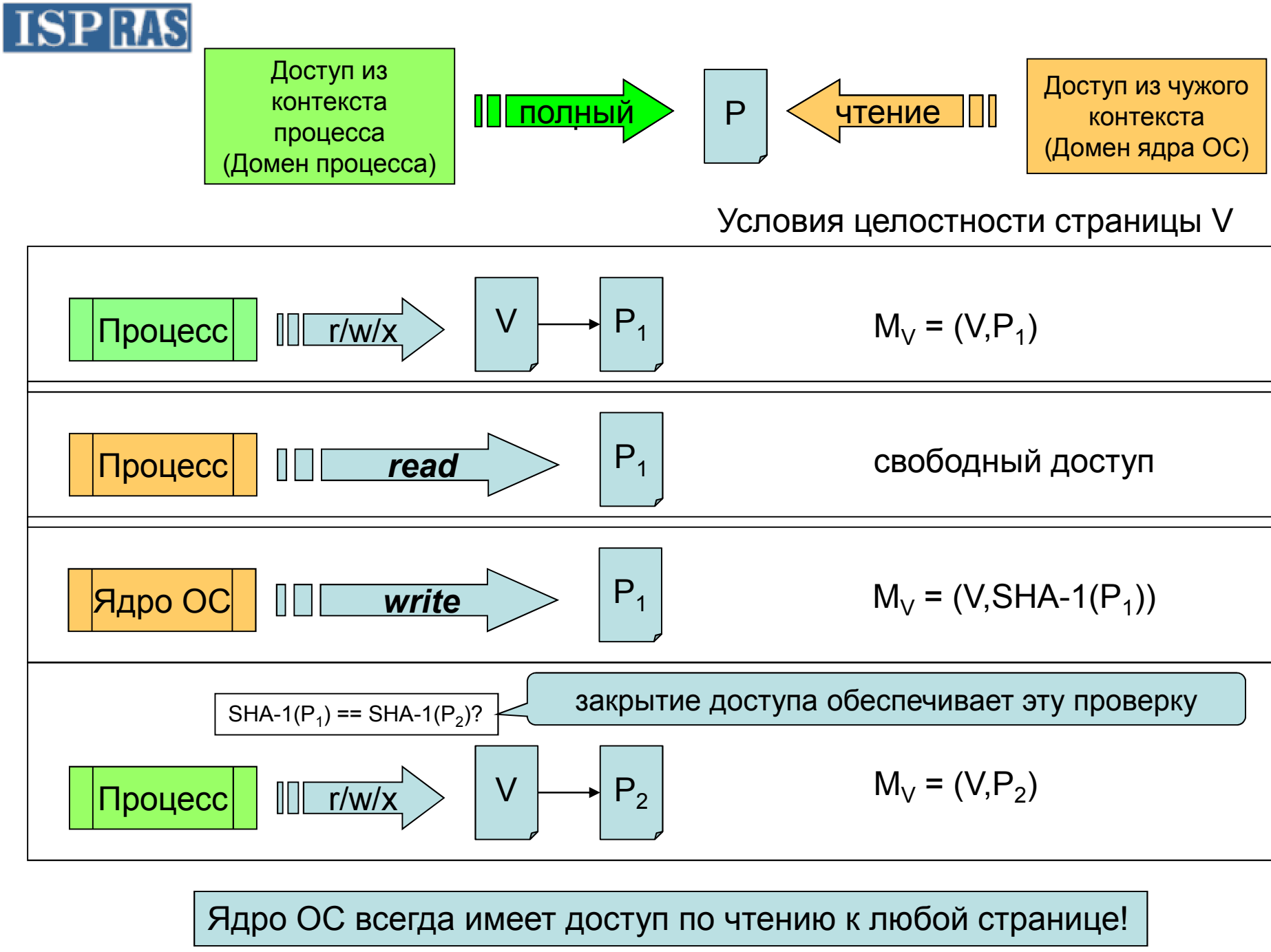
- Переключение доменов производится гипервизором, когда виртуальная машина *не выполняется*

- Домены взаимосвязаны и синхронизируются в момент доступа к странице с действием, нарушающим ограничения (исключение #NPF)

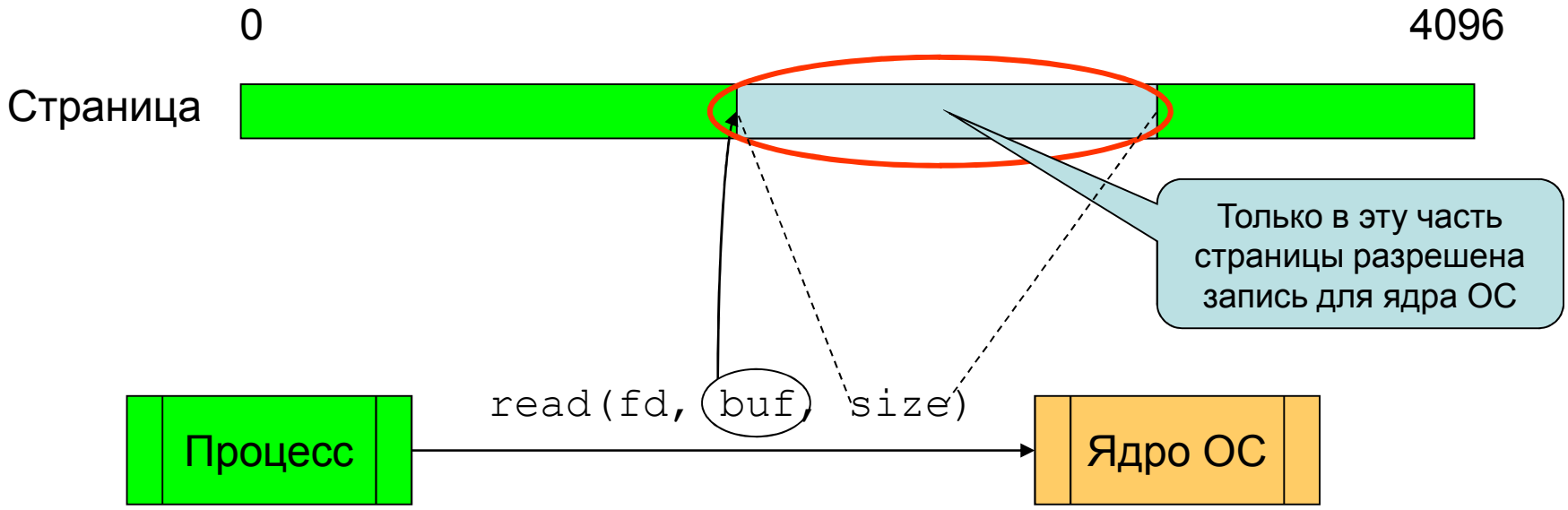
- Полный доступ к странице открыт не более чем в одном домене

Защита физической памяти




Доступ из контекста процесса (Домен процесса)
полный
P
чтение
Доступ из чужого контекста (Домен ядра ОС)
Условия целостности страницы V
Процесс
r/w/x
V
P1
MV = (V,P1)
Процесс
read
P1
свободный доступ
Ядро ОС
write
P1
MV = (V,SHA-1(P1))
SHA-1(P1) == SHA-1(P2)?
закрытие доступа обеспечивает эту проверку
Процесс
r/w/x
V
P2
MV = (V,P2)
Ядро ОС всегда имеет доступ по чтению к любой странице!

OUT-параметры



- Перед передачей управления ОС специальный модуль в ядре выделяет «теневой» буфер в памяти процесса и передает его адрес ядру ОС (вместо исходного адреса)
- После возврата из системного вызова гипервизор копирует данные из теневого буфера в исходный
- Не подходит для «асинхронных» операций модификации

ISP RAS

Открытые вопросы

- Память, разделяемая несколькими доверенными процессами
  - IPC
  - потоки
  - mmap файлы
- Асинхронные операции записи
  - в частности необходимы для поддержки много-потоковых приложений
- Модификация содержимого виртуальной страницы посредством изменения ее трансляции
  - также позволит избавиться от libdisasm
- Нарушение целостности процесса за счет некорректного поведения системных вызовов
  - например, нарушение примитивов синхронизации
- Другие операционные системы
  - Windows

ISP RAS

Итоги

- Система защиты выполнения пользовательского процесса
  - Основана на контроле целостности потока управления и адресного пространства процесса
  - Не рассчитывает на корректность работы операционной системы
- Основное предназначение – ограничение доступа системного кода к аппаратным ресурсам с сохранением доступа пользовательским приложениям
  - Может использоваться для (независимого) усиления системы безопасности ОС
  - Может использоваться для ограничения доступа процессов к ресурсам (на уровне системных вызовов)
- Прозрачная работа по отношению к ОС и пользовательским приложениям
  - не требует модификации ОС
  - не требует адаптации приложений
- Типовая аппаратура
  - процессор с поддержкой аппаратной виртуализации
- Целиком основана на возможностях аппаратуры
  - не требует бинарной трансляции и интерпретации кода приложения/ОС

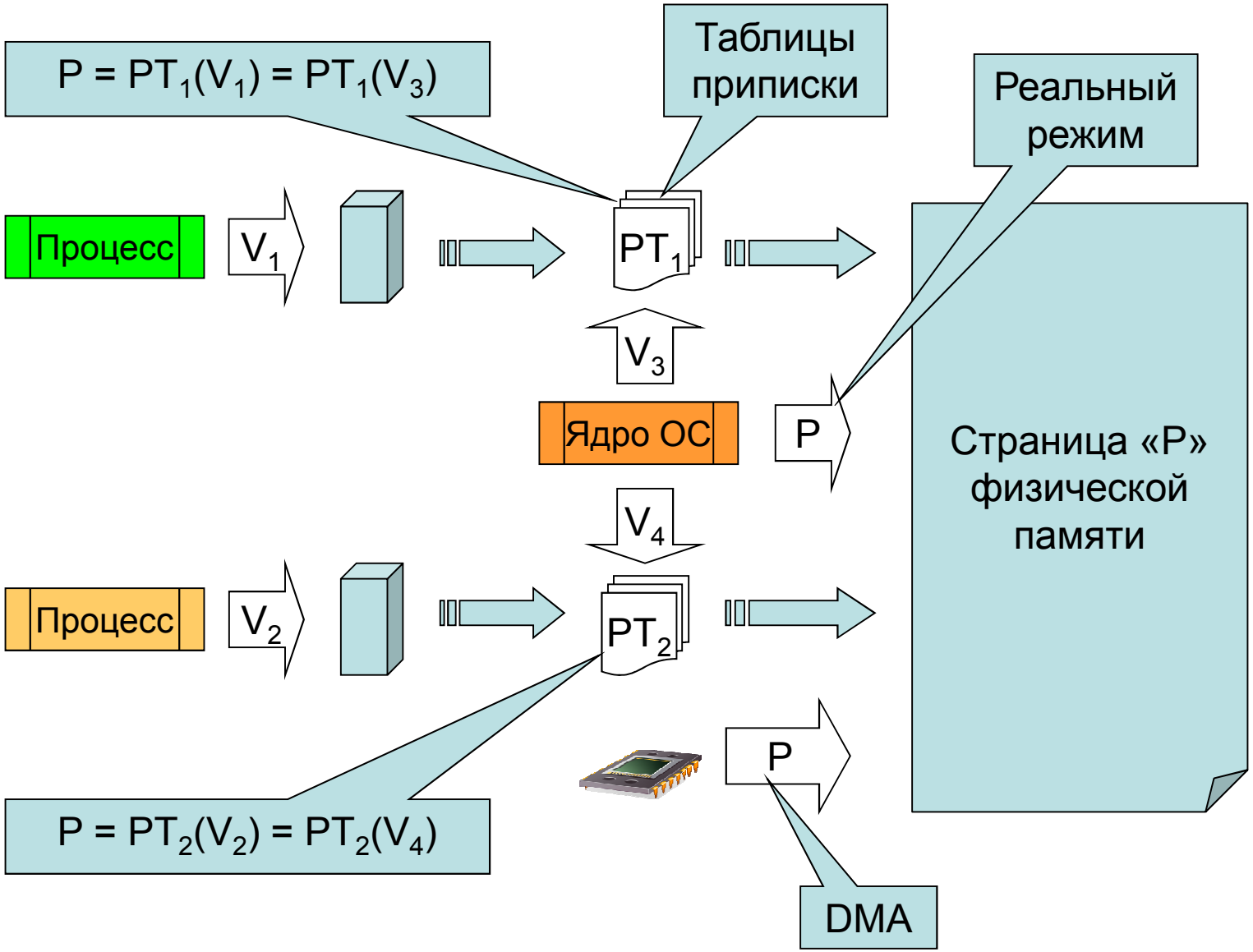
ISP RAS

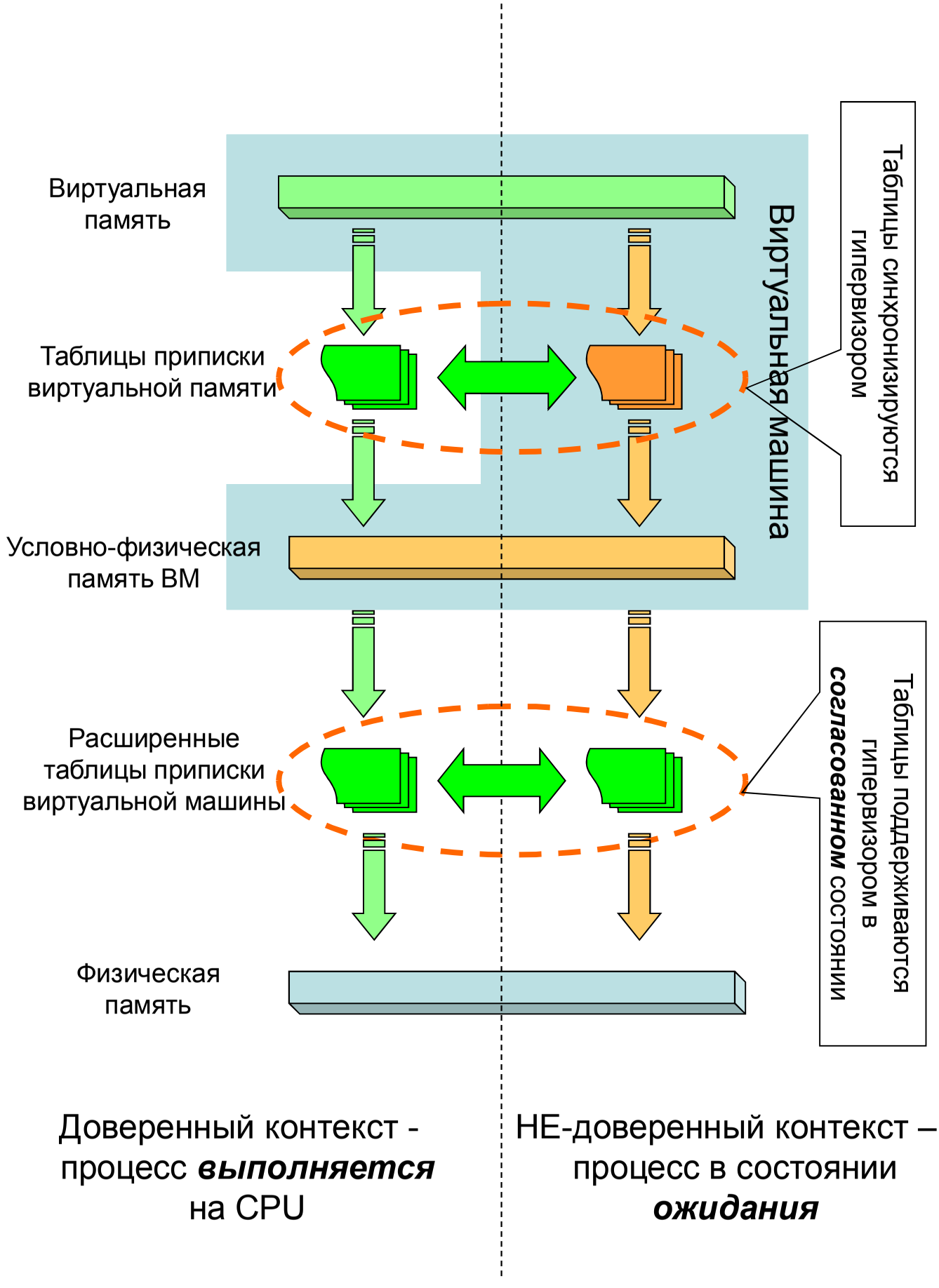
Вопросы?

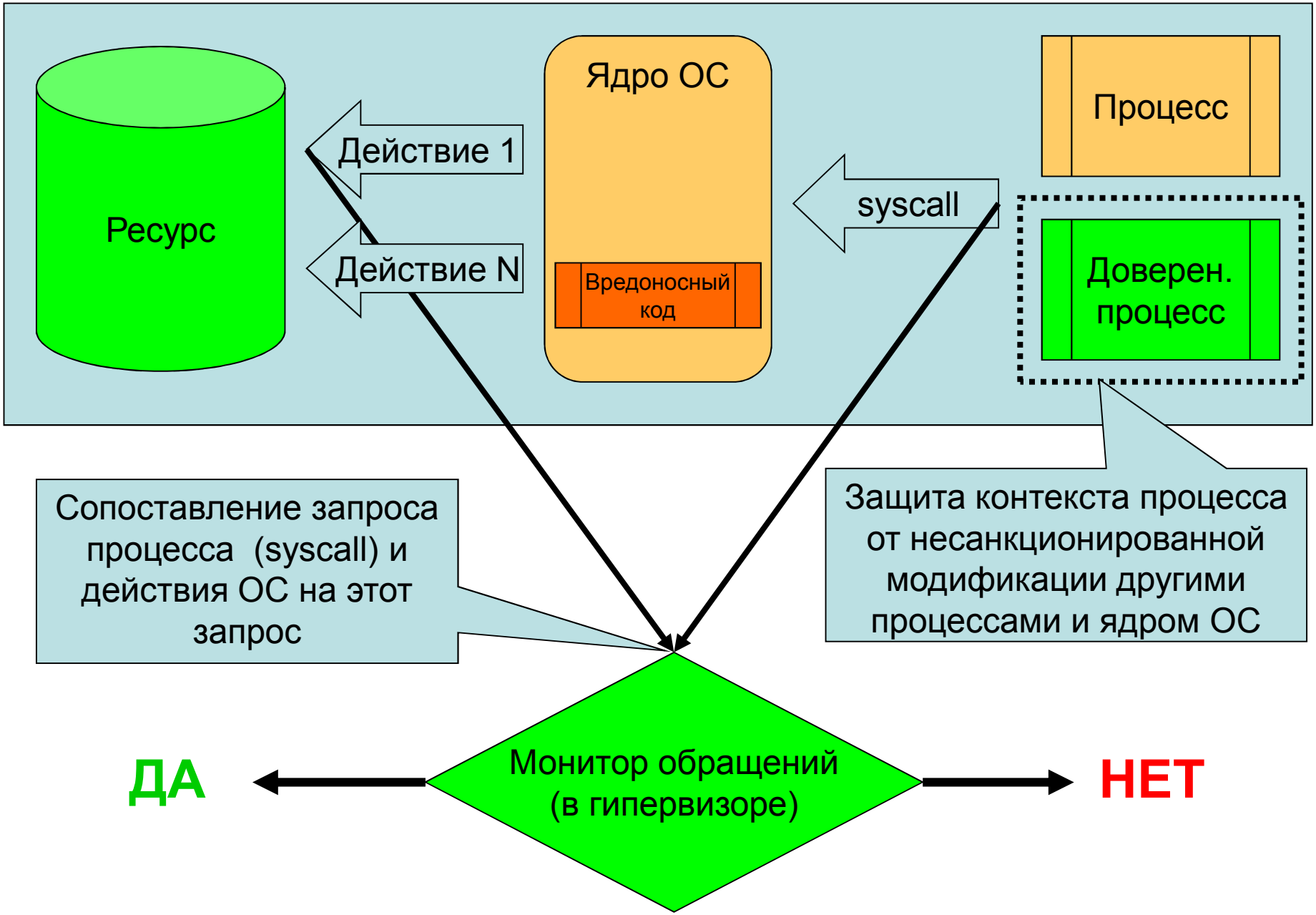
ISP RAS

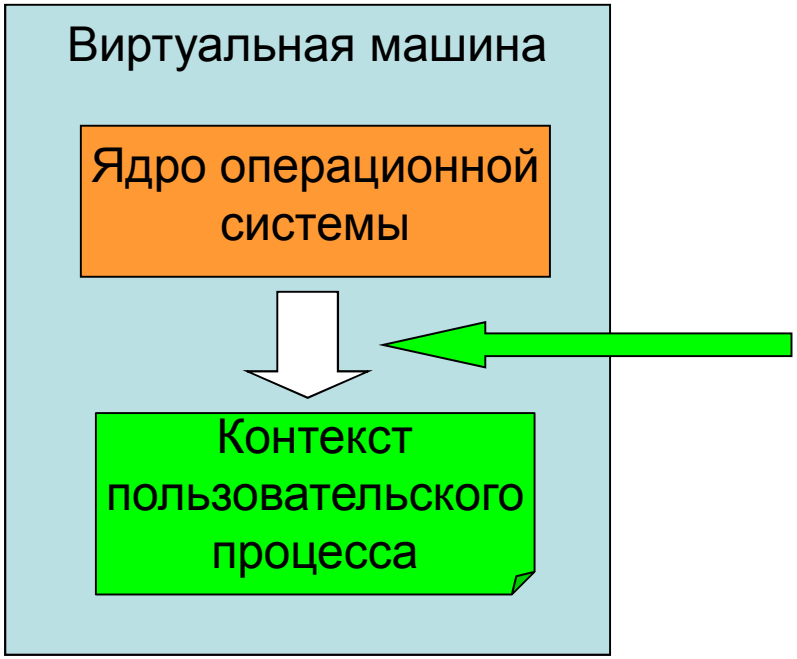
- Типовые комплектующие (процессор)
  - с поддержкой аппаратной виртуализации
- Целиком опирается на возможности аппаратуры
  - не требует бинарной трансляции и интерпретации кода приложения/ОС
- Пользовательские приложения и операционная система не требуют адаптации
  - требуется поддержка загрузки модулей в ядро Linux
- Компоненты системы располагаются как в гипервизоре, так и внутри виртуальной машины
  - цель – минимизация кода в теле гипервизора
  - абстрагирование модулей в гипервизоре от конкретных структур данных ОС
- Типовые комплектующие (процессор)
  - с поддержкой аппаратной виртуализации
- Целиком опирается на возможности аппаратуры
  - не требует бинарной трансляции и интерпретации кода приложения/ОС

Варианты доступа к странице физической памяти с адресом «P»



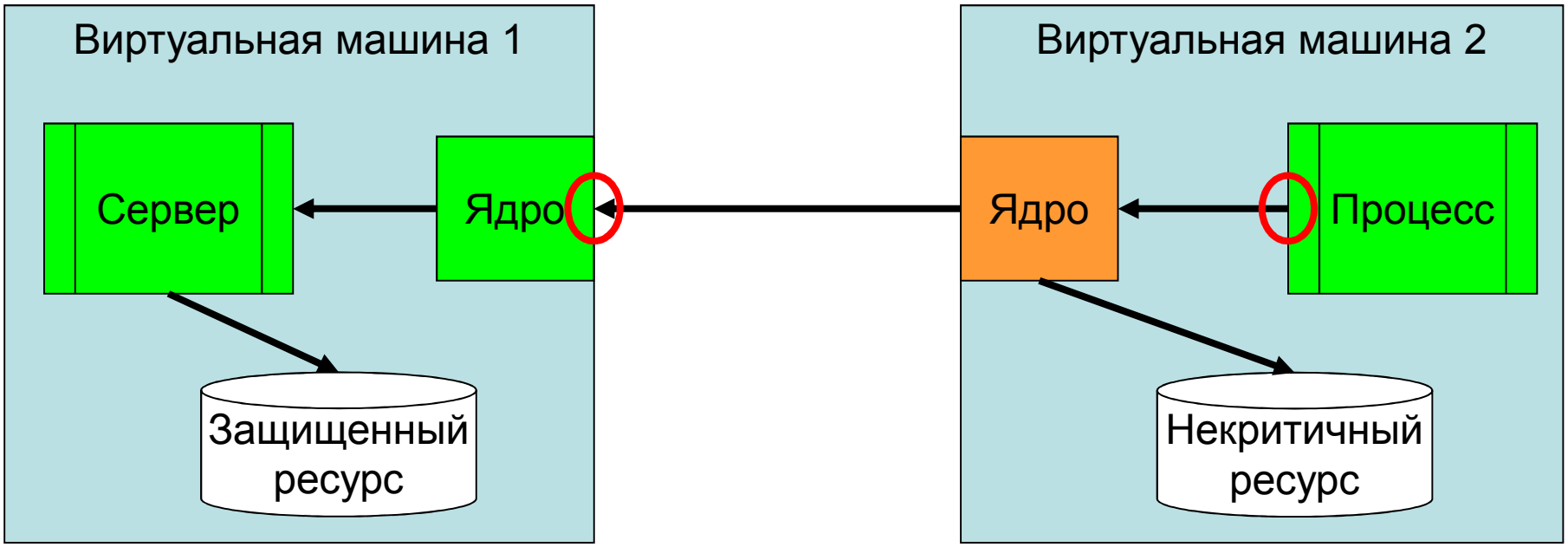

Виртуальная память
Таблицы приписки виртуальной памяти
Условно-физическая память ВМ
Виртуальная машина
Таблицы синхронизируются гипервизором
Расширенные таблицы приписки виртуальной машины
Физическая память
Таблицы поддерживаются гипервизором в согласованном состоянии
Доверенный контекст - процесс выполняется на CPU
НЕ-доверенный контекст – процесс в состоянии ожидания

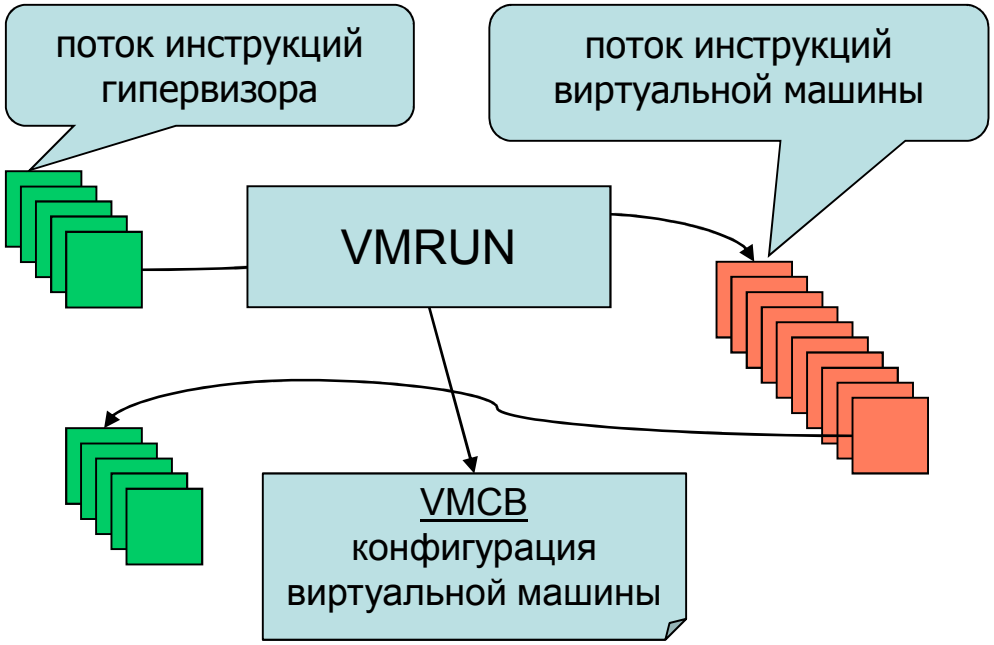

ISP RAS
Ядро ОС
Процесс
Ресурс
Действие 1
Действие N
syscall
Вредоносный код
Доверен. процесс
Сопоставление запроса процесса (syscall) и действия ОС на этот запрос
Защита контекста процесса от несанкционированной модификации другими процессами и ядром ОС
ДА
Монитор обращений (в гипервизоре)
НЕТ

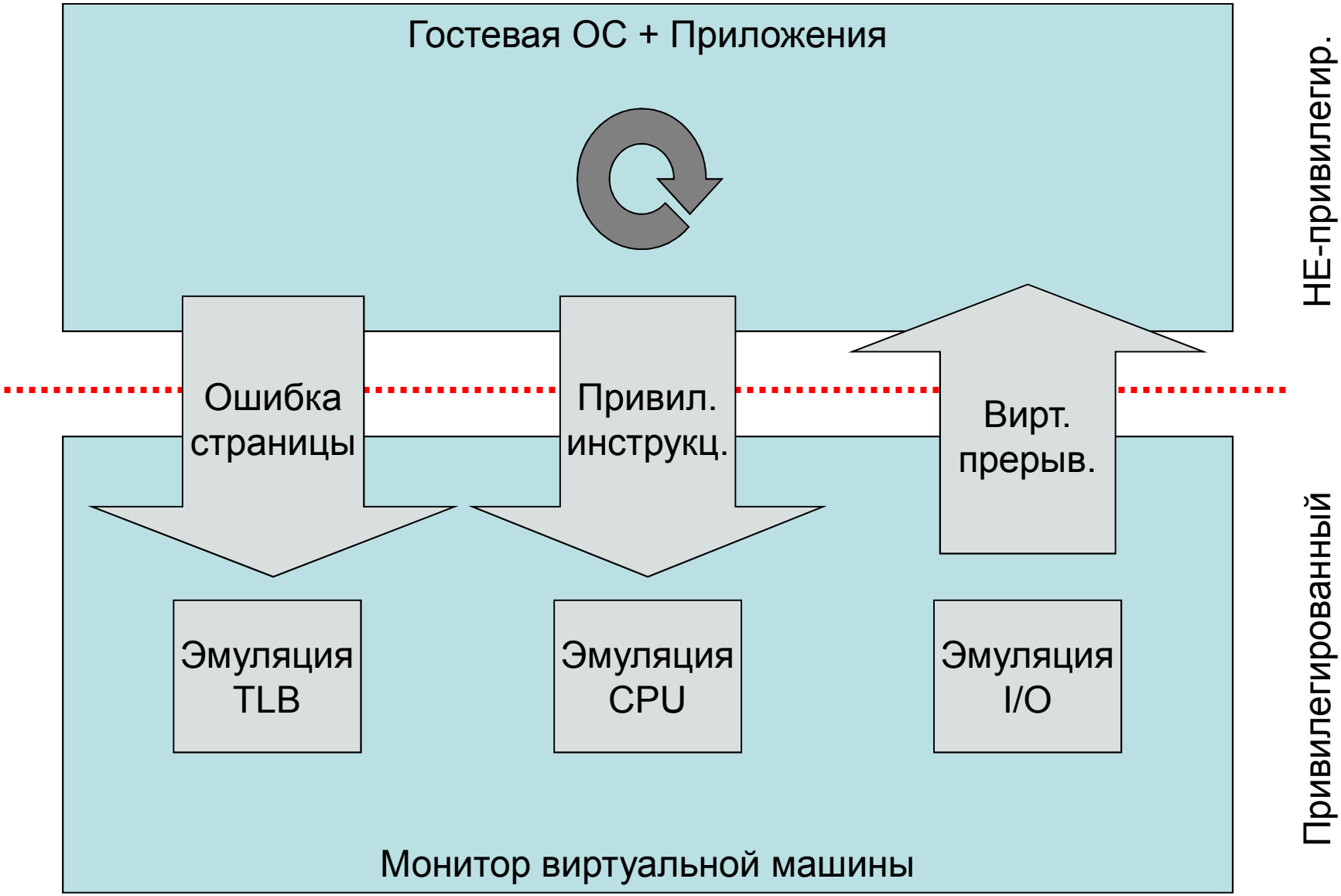


<u>Контроль целостности</u>:

- типовая ОС
- ОС и приложение не адаптируются
- системные вызовы
- вытеснение виртуальной памяти
- DMA операции


Виртуальная машина 1
Сервер
Ядро
Защищенный ресурс
Виртуальная машина 2
Ядро
Процесс
Некритичный ресурс


поток инструкций гипервизора
поток инструкций виртуальной машины
VMRUN
VMCB
конфигурация виртуальной машины


Гостевая ОС + Приложения
Ошибка страницы
Привил. инструкц.
Вирт. прерыв.
Эмуляция TLB
Эмуляция CPU
Эмуляция I/O
Монитор виртуальной машины
НЕ-привилегир.
Привилегированный