

НИИ системных исследований РАН

# ОСРВ на базе стандарта ARINC 653

архитектура и управление ресурсами

Райко Г.О., Рогожкин С.А., Павлов А.Н.

[linux@niisi.msk.ru](mailto:linux@niisi.msk.ru)

# ARINC Specification 653

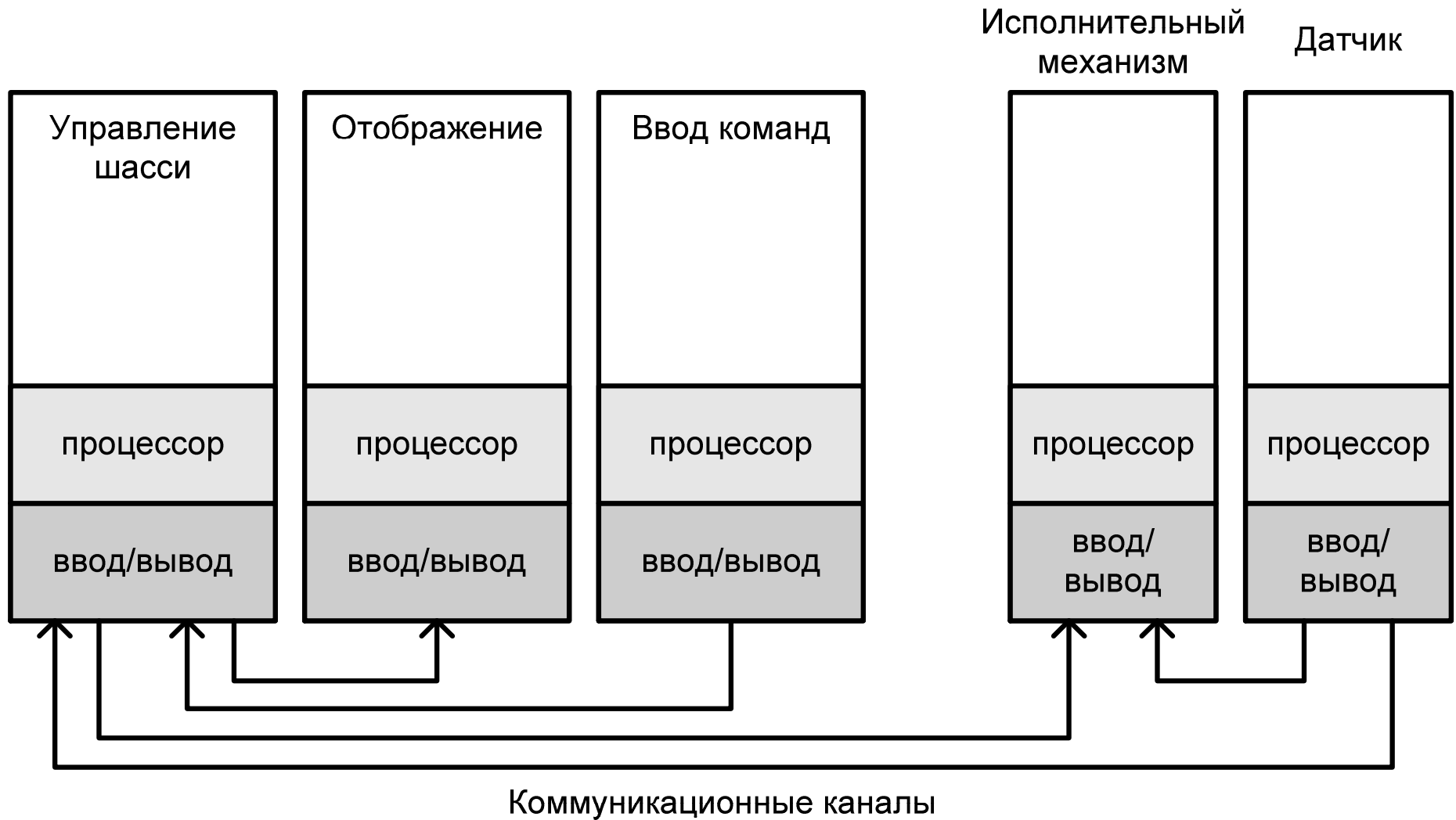
## Avionics Application Software Standard Interface

- Part 1 - Required Services  
ARINC Specification 653P1-2 (03.2006)
- Part 2 - Extended Services  
ARINC Specification 653P2-1 (12.2008)
- Part 3 - Conformity Test Specification  
ARINC Specification 653P3 (10.2006)

AEEC – Airlines Electronic Engineering Committee

ARINC – Aeronautical Radio, Incorporated

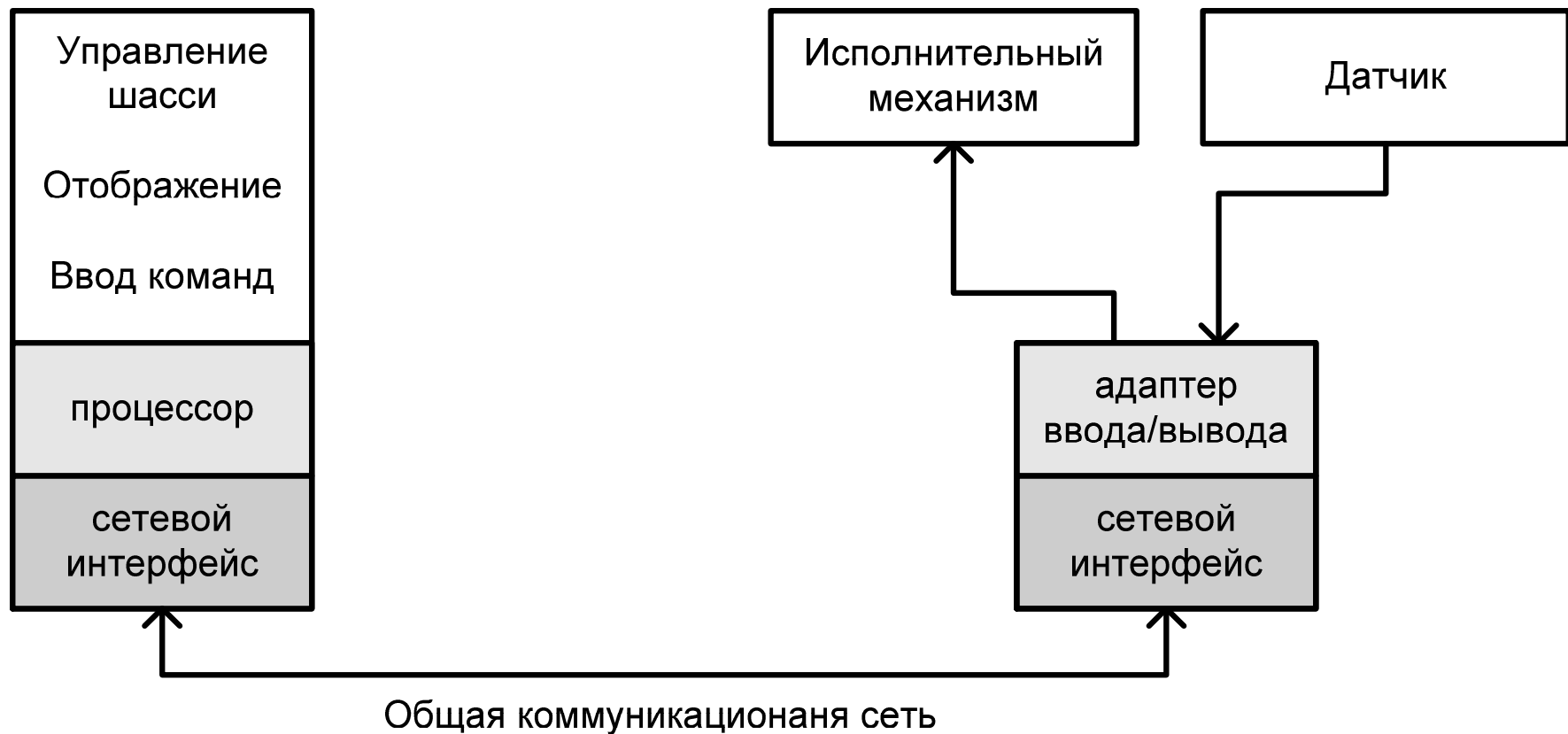
# Федеративная архитектура



«Поскольку работа автомата тяги не согласована с работой автопилота, контроль скорости, выполняемый автоматом тяги, постоянно нарушает работу автопилота по управлению курсом и наоборот. При полетах в условиях турбулентности или при сдвигах ветра это проявляется настолько ярко, что качество полета становится неприемлемым. Широко применявшийся способ борьбы с указанной проблемой – перевод автопилота в режим Удержание высоты (например, на старых Боингах 747-200/300). На новых самолетах проблему удалось свести к приемлемому уровню после введения обратной связи в автомат тяги, применения методов компенсации энергии, компенсации турбулентности и нелинейных методов обнаружения/компенсации сдвигов ветра, процесс разработки которых был очень сложным и дорогим.»

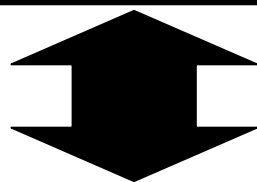
«Из-за отсутствия координации управления, режимы автопилота Выбор высоты и Вертикальная скорость никогда не функционировали удовлетворительно... это привело к разработке нового режима Смена эшелона, впервые примененного в Боингах 757/767... однако логика работы автопилота в этом режиме строится на некоторых допущениях, которые верны только при определенных условиях... в результате были зарегистрированы случаи, когда в указанном режиме команды пилота выполняются неправильно.»

# Integrated Modular Avionics (IMA)



# Критерий соответствия

Система разделов должна предоставить такое же ограничение последствий неисправностей, что и идеализированная система, в которой каждый раздел находится на отдельном процессоре, а коммуникация между разделами обеспечивается выделенными каналами.

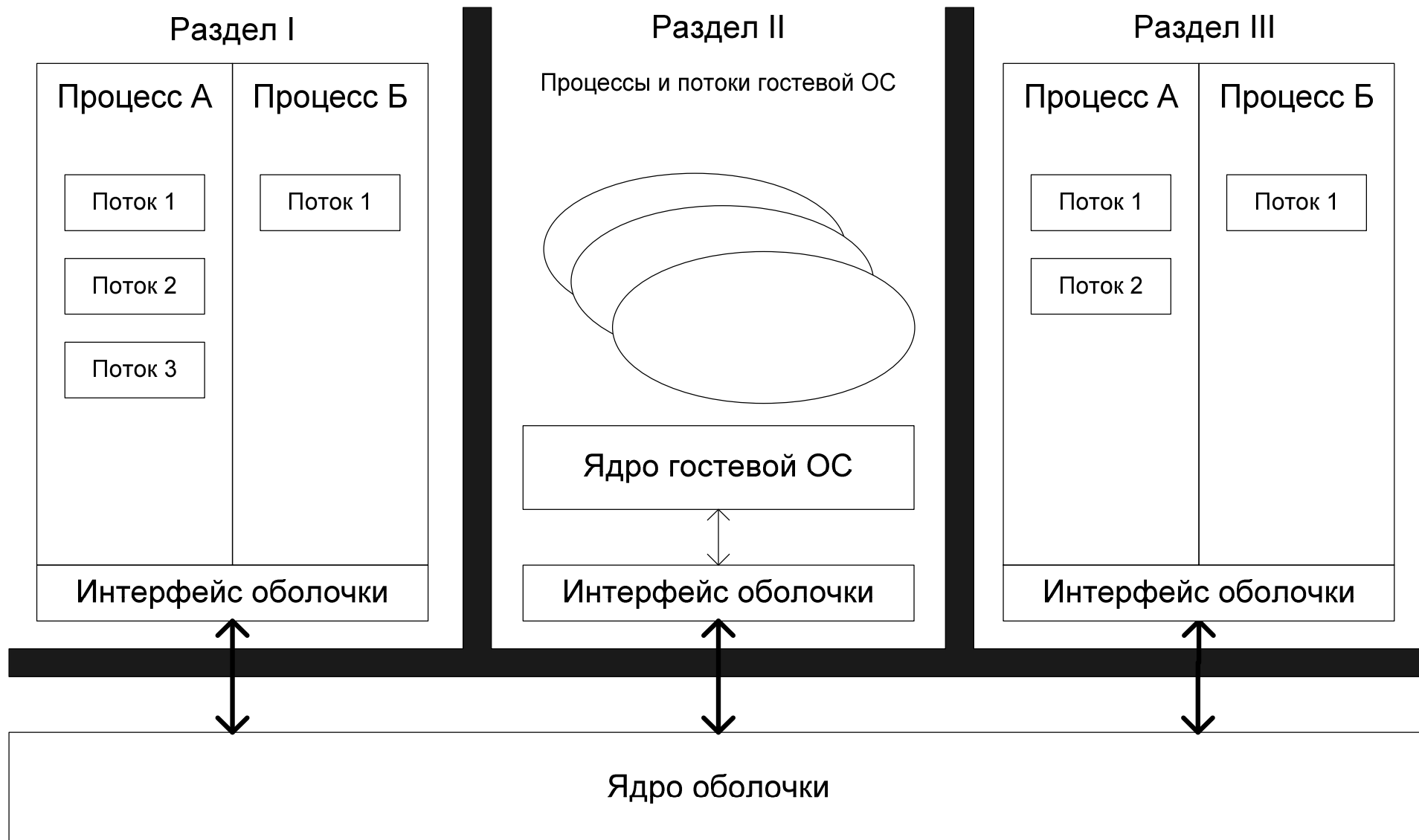


На поведение и исполнение программы в одном разделе не должны влиять программы из остальных разделов

# Разделение ресурсов

- Пространственное разделение
- Временное разделение

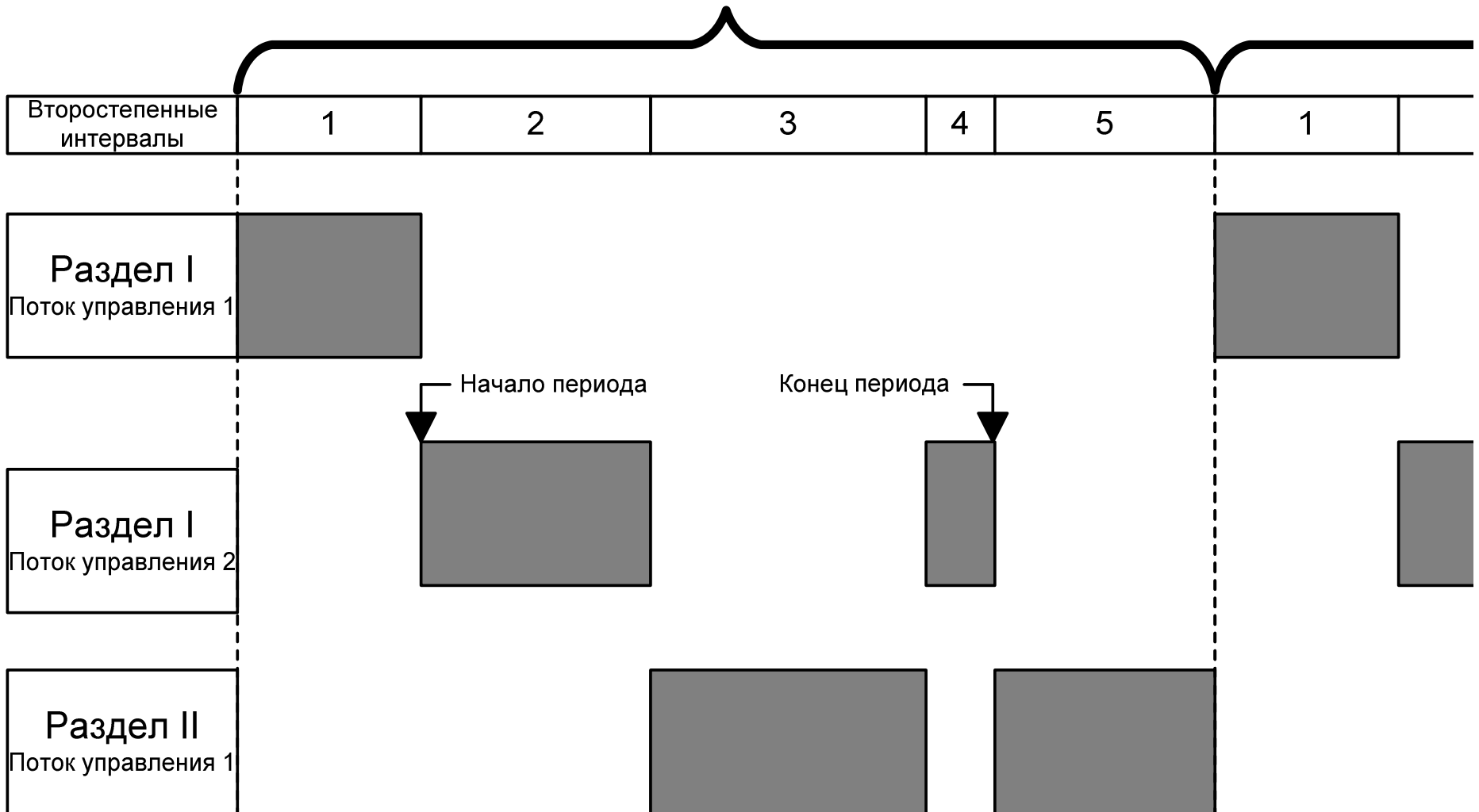
# Разделы, процессы, потоки





# Планирование

Основной интервал



# ARINC 653 изнутри

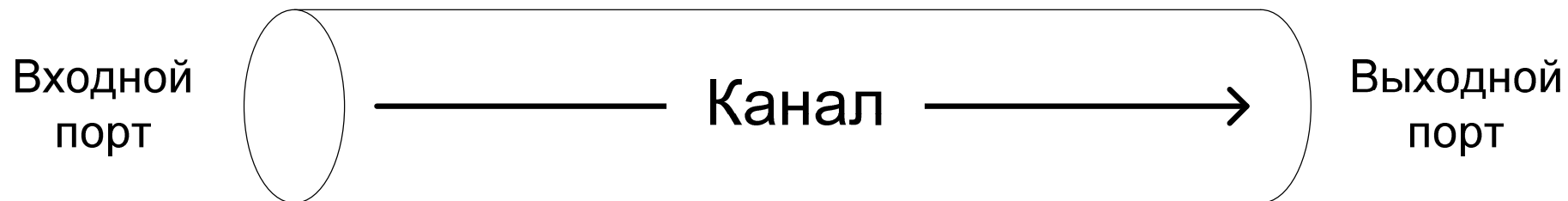
- APEX – Application/Executive  
около 50 системных вызовов
- Статическое распределение  
ресурсов
- Вызовы для управления памятью  
отсутствуют

## 2.3.4 Memory Allocation

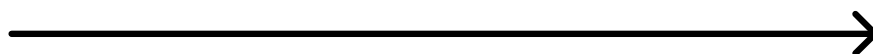
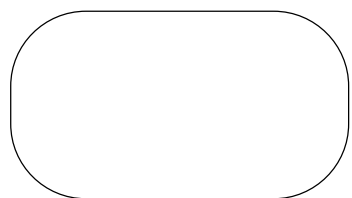
Partitions, and therefore their associated memory spaces, are defined during system configuration.

There are no memory allocation services in the APEX interface.

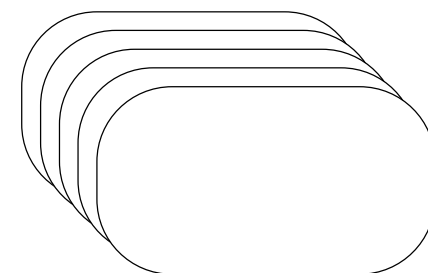
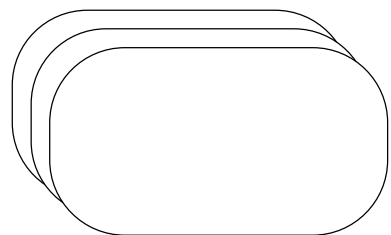
# Коммуникации между разделами



Sampling mode

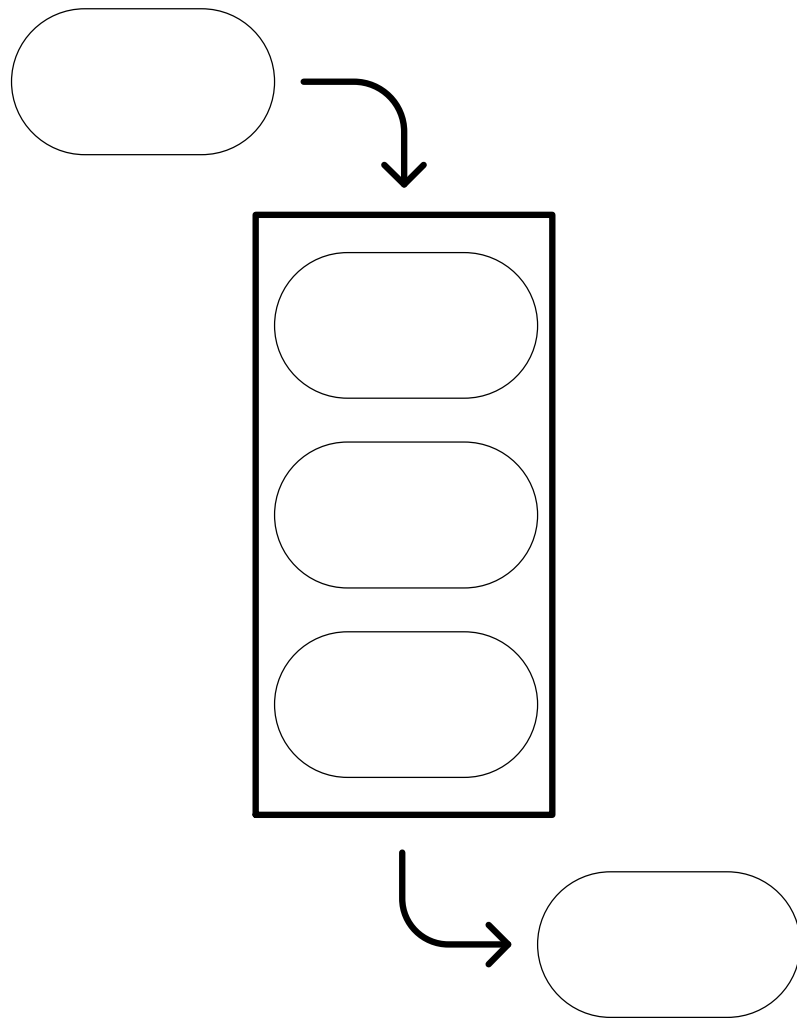


Queuing mode

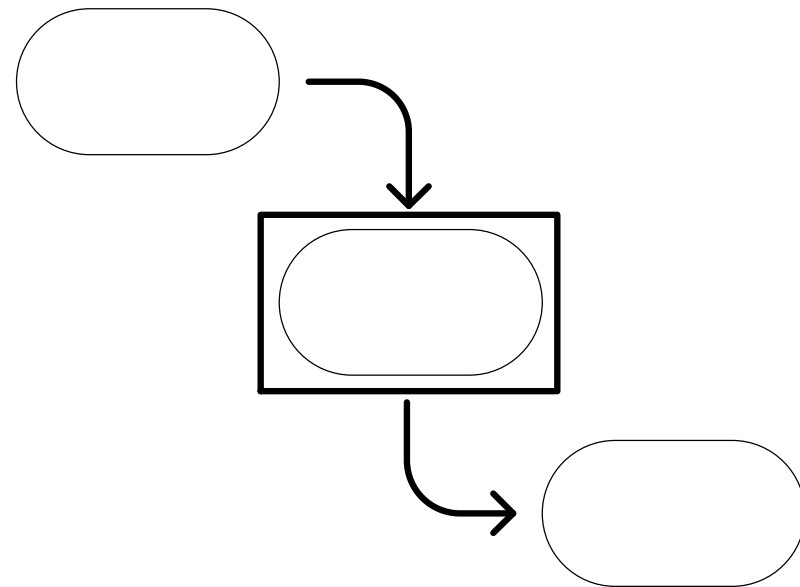


# Коммуникации внутри разделов

Буфер



Доска объявлений



Семафор

Событие

# Другие стандарты

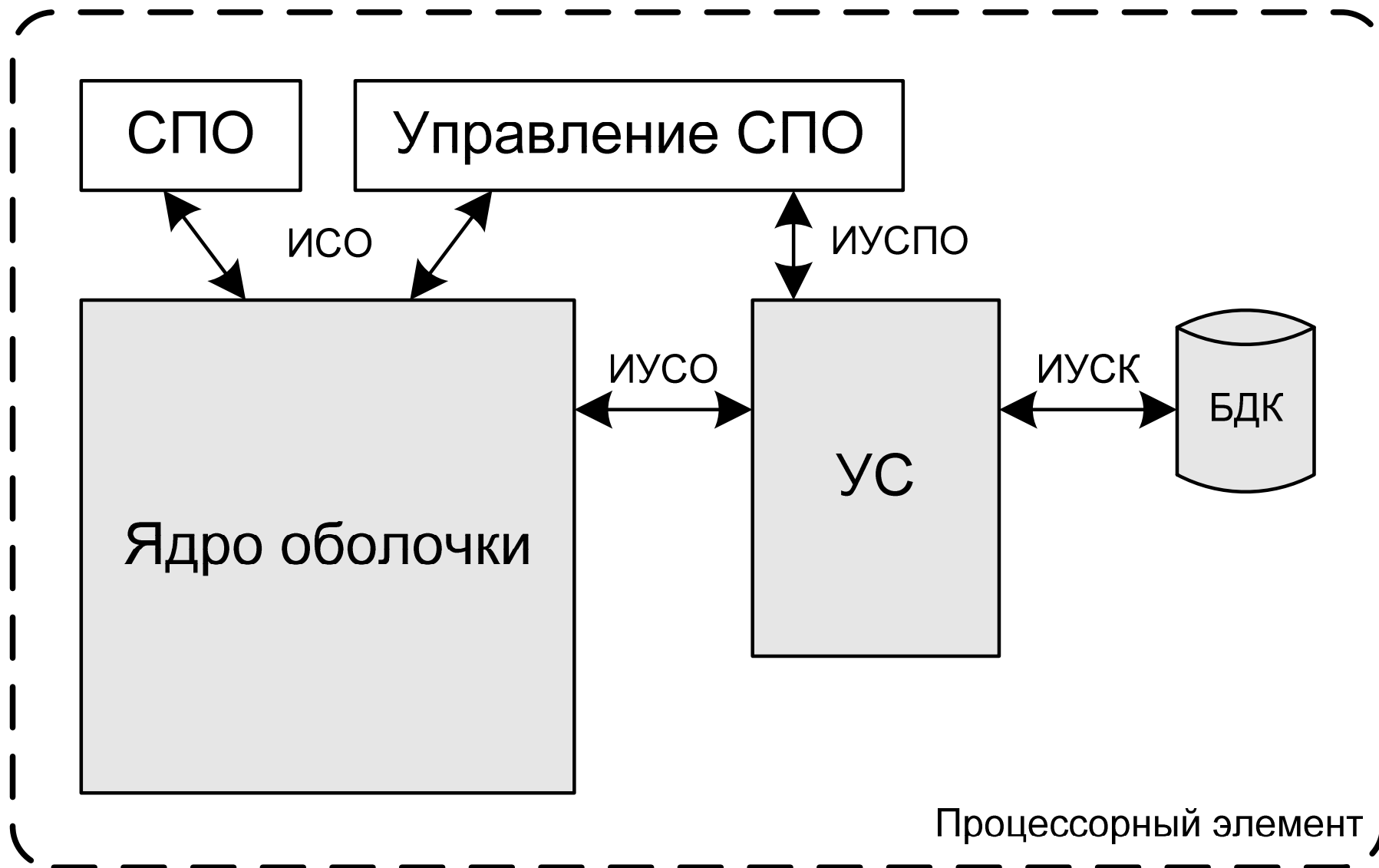
- **Программа ASAAC**  
Allied Standards Avionics Architecture Council
  - Def Stan 00-74
  - STANAG 4626
  
- **AUTOSAR**  
Automotive Open System Architecture

# Компоненты системы

- Ядро оболочки
  - Служба управления системой
  - База данных конфигурации
- 

- Служба управления СПО
- СПО

# Процессорный элемент

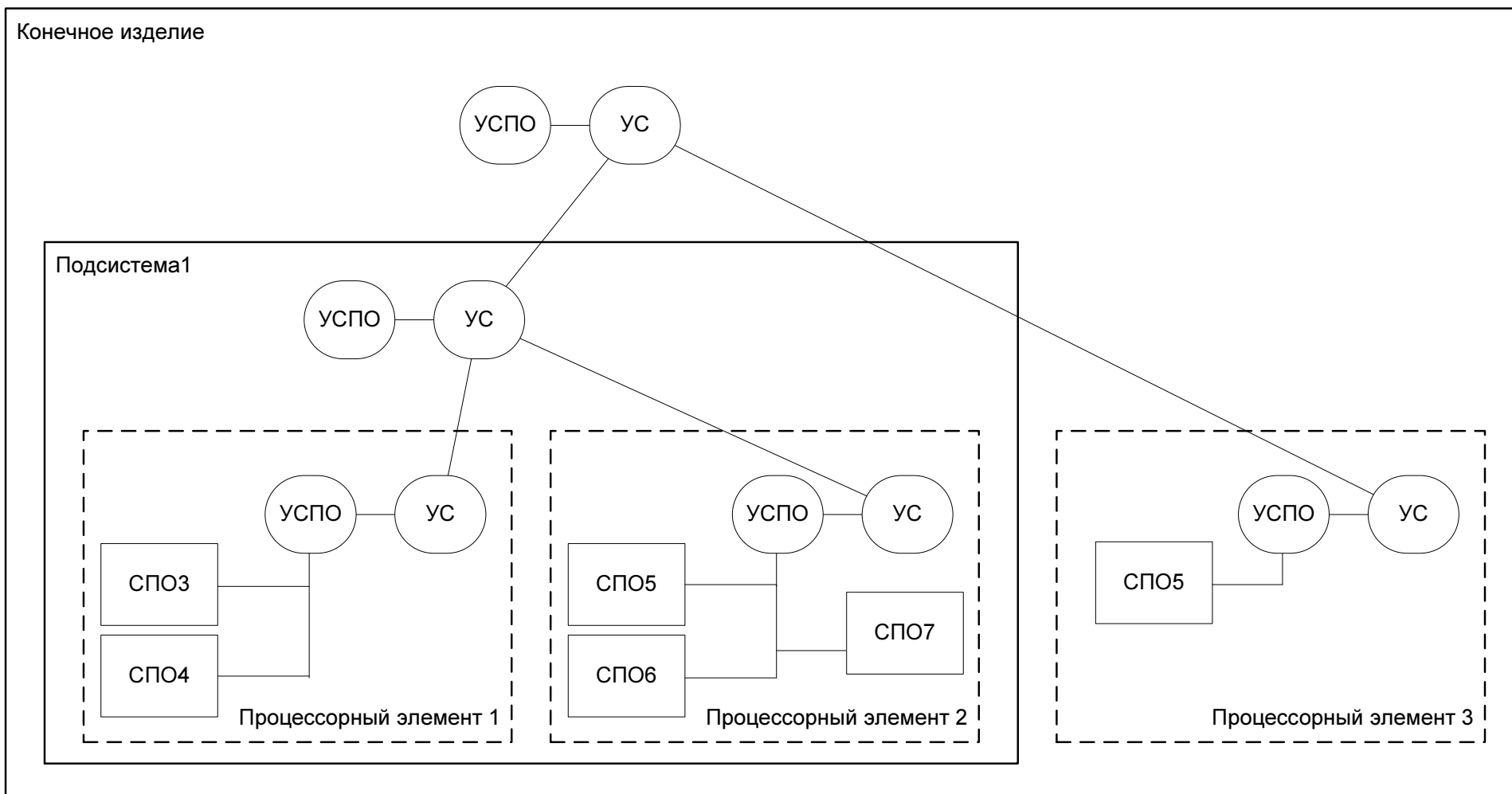


# Многоуровневая структура

- Конечное изделие (КИ)
- Подсистема
- Процессорный элемент



# Иерархия управления изделием



# Разработка СПО

- Разработка СПО – сложный итеративный процесс в который вовлечены несколько коллективов разработчиков
- При разработке используются формальные описания для КИ
- За разработку СПО в целом ответственен главный конструктор изделия (отдел главного конструктора)

# Разработка: исходные данные

- Формализованное описание вычислительной задачи
- Формализованное описание аппаратуры вычислительного комплекса

# Разработка: результаты

- Привязка программных компонент к процессорам
- Использование ресурсов коммуникационной среды

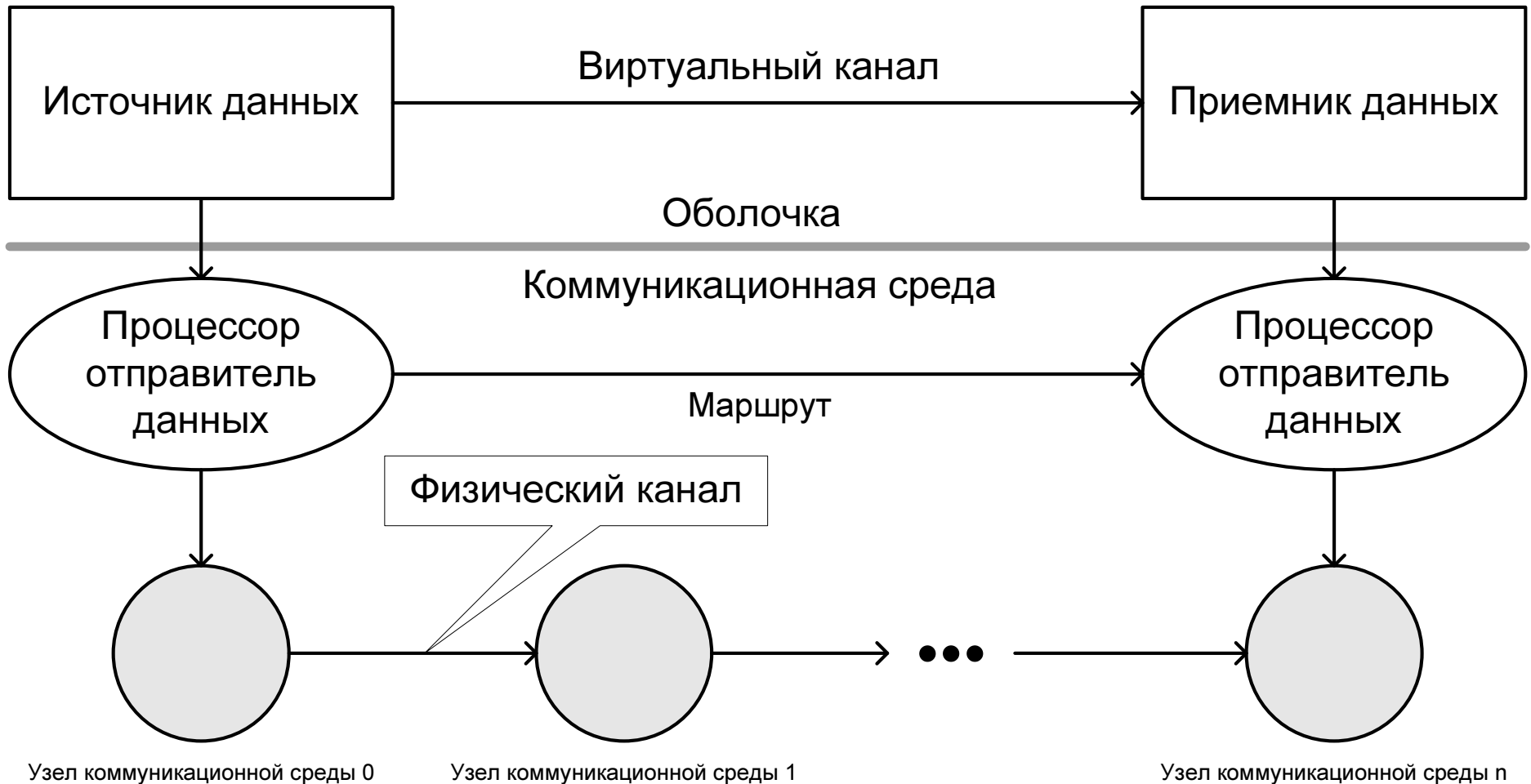
# Разработка: план действий

- Составляем формальное описание задачи
- Определяем требуемое число процессоров, выбираем подходящую конфигурацию аппаратуры (из предлагаемых)
- Составляем формальное описание аппаратуры и ПО на языке XML
- Валидируем составленные описания

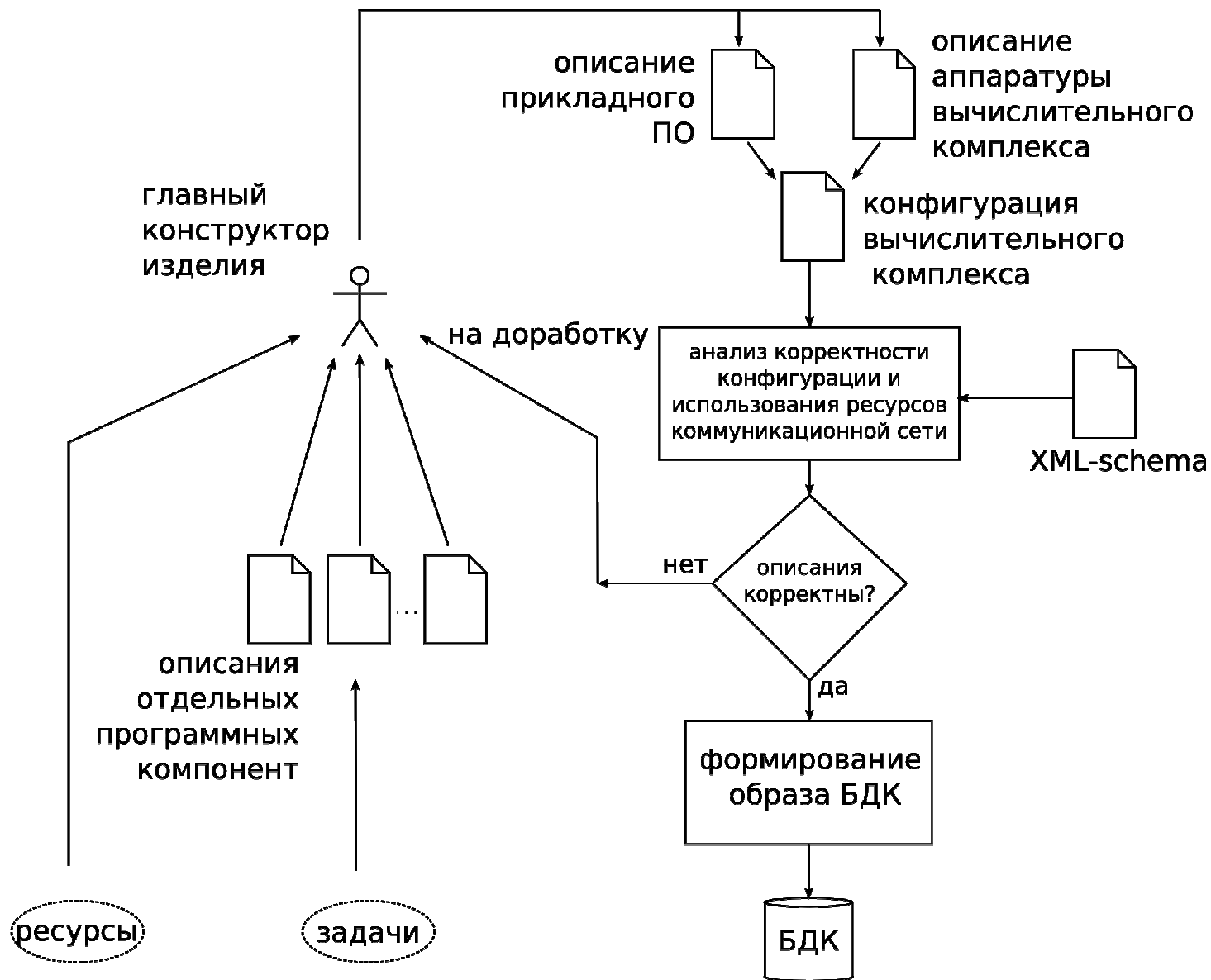
# Разработка: формальные модели

- При разработке используются формальные описания:
  - конфигурация СПО
  - конфигурация аппаратуры
- Разработано ПО для валидации различных конфигураций

# Валидация коммуникаций

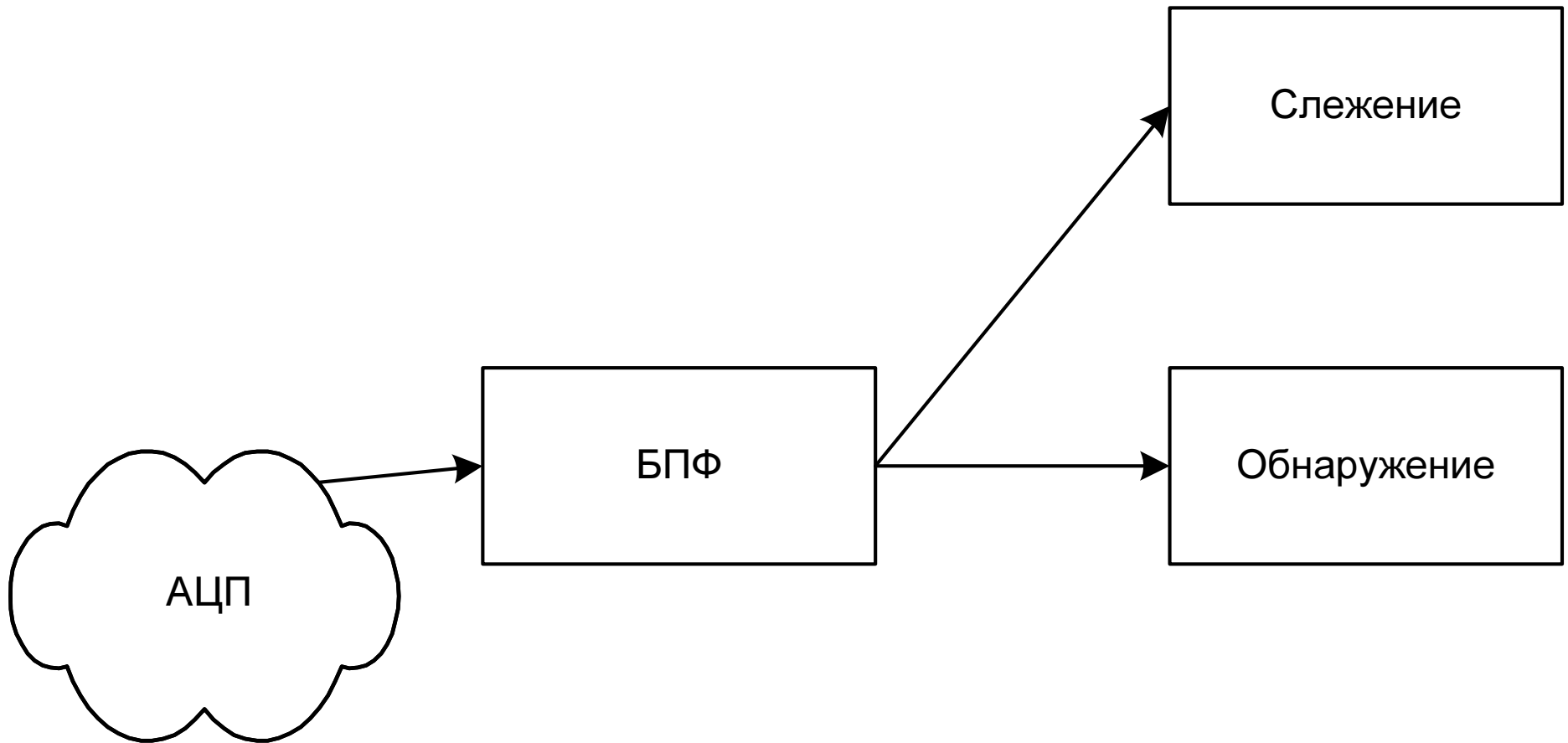


# Схема процесса разработки СПО





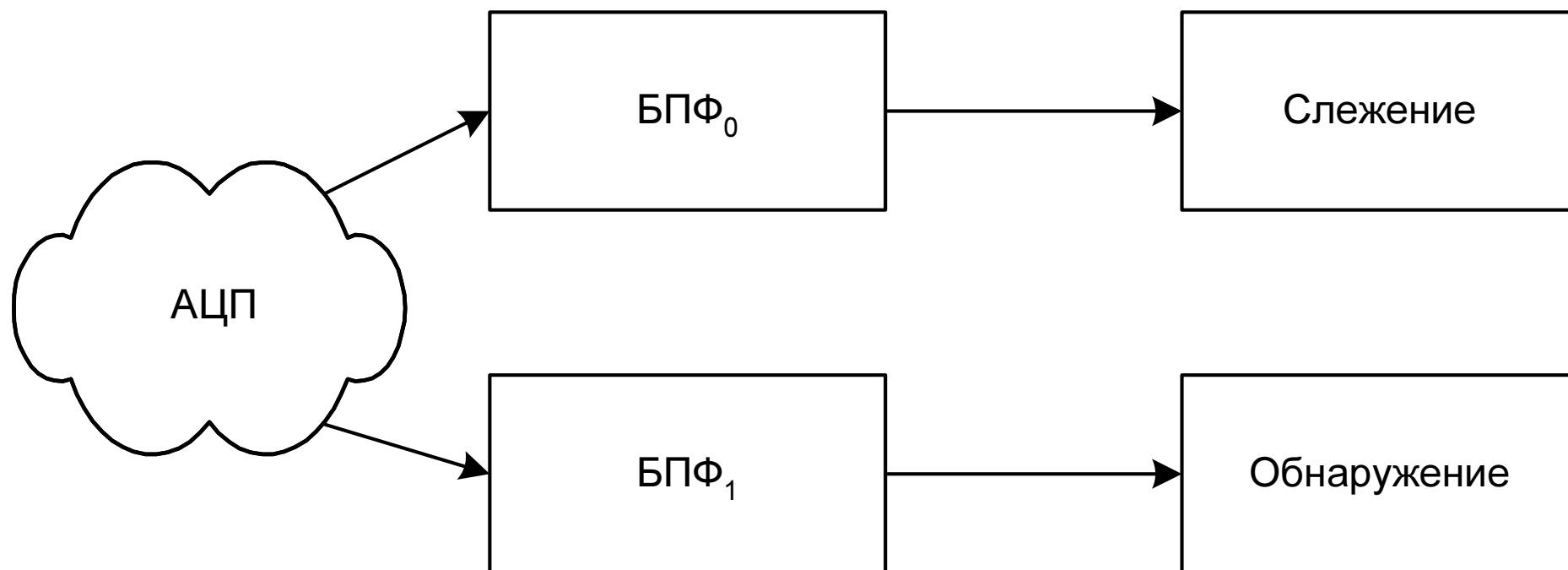
# Пример: задача ЦОС



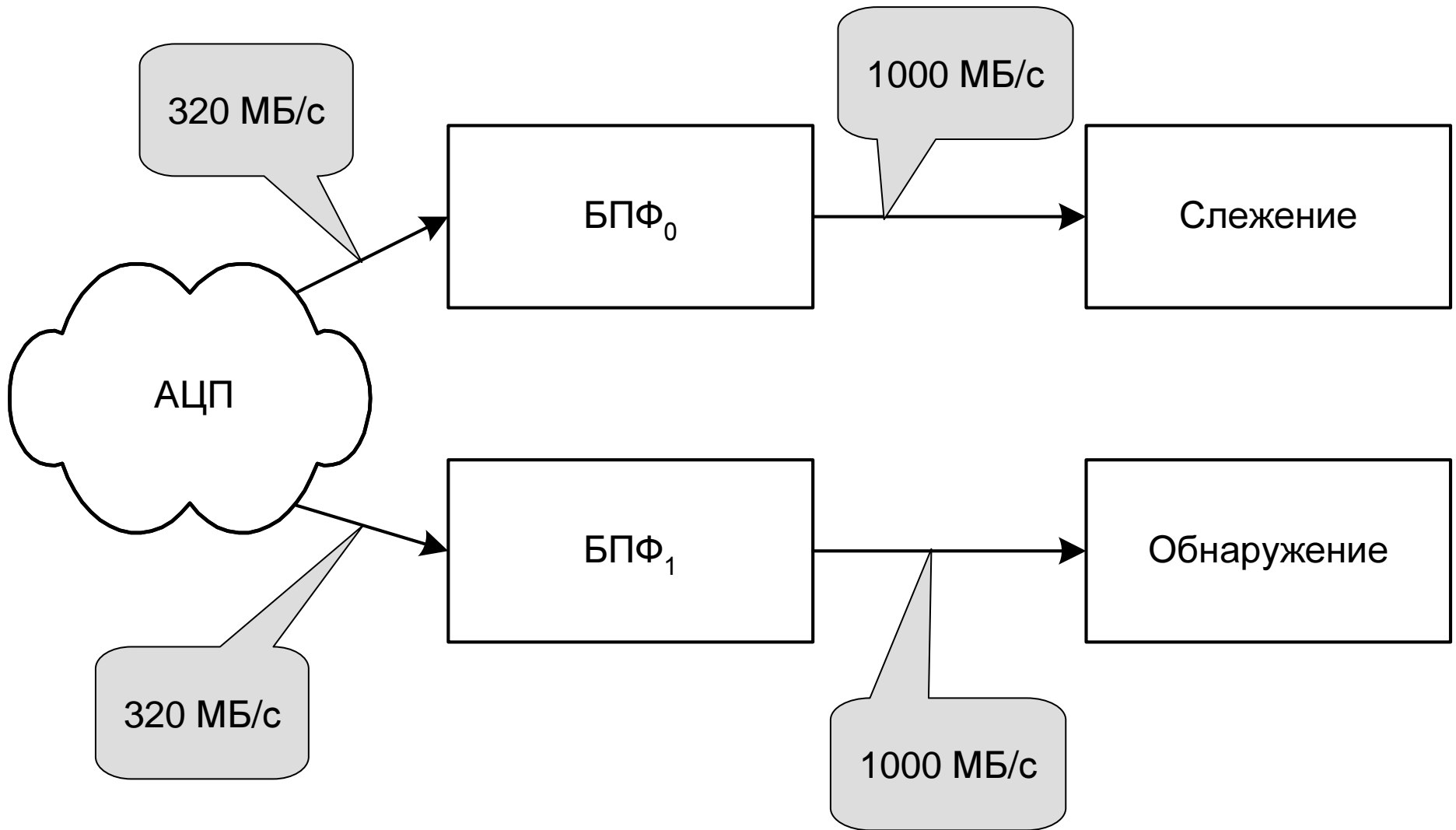
# Пример: аппаратура

- Микропроцессоры с архитектурой MIPS и расширениями для обработки сигналов
  - 10 ГФлопс (32-разрядные числа)
- Коммуникационная среда RapidIO
  - последовательные каналы
  - параллельные каналы
  - 500 + 500 МБ/с

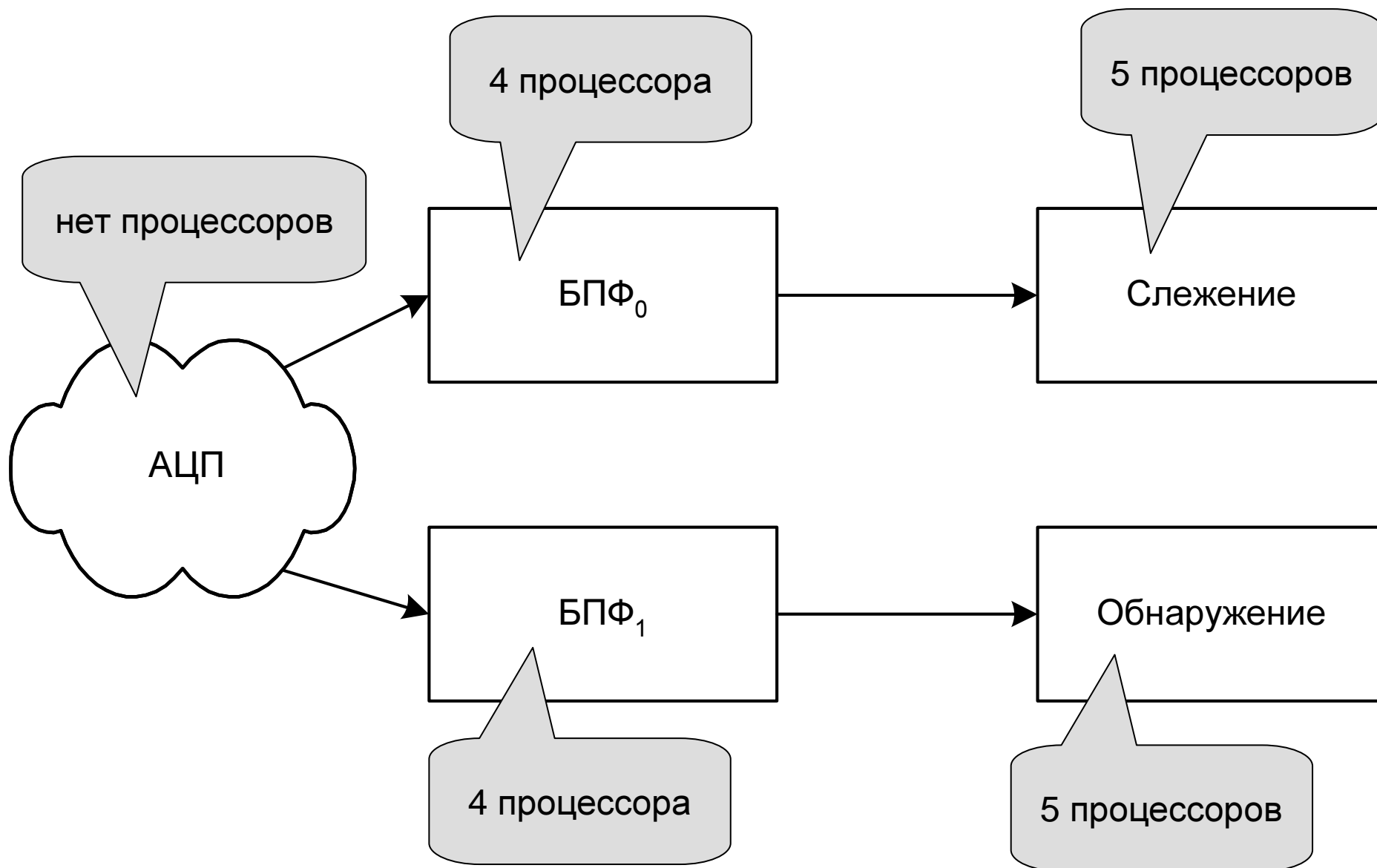
# Пример: схема решения



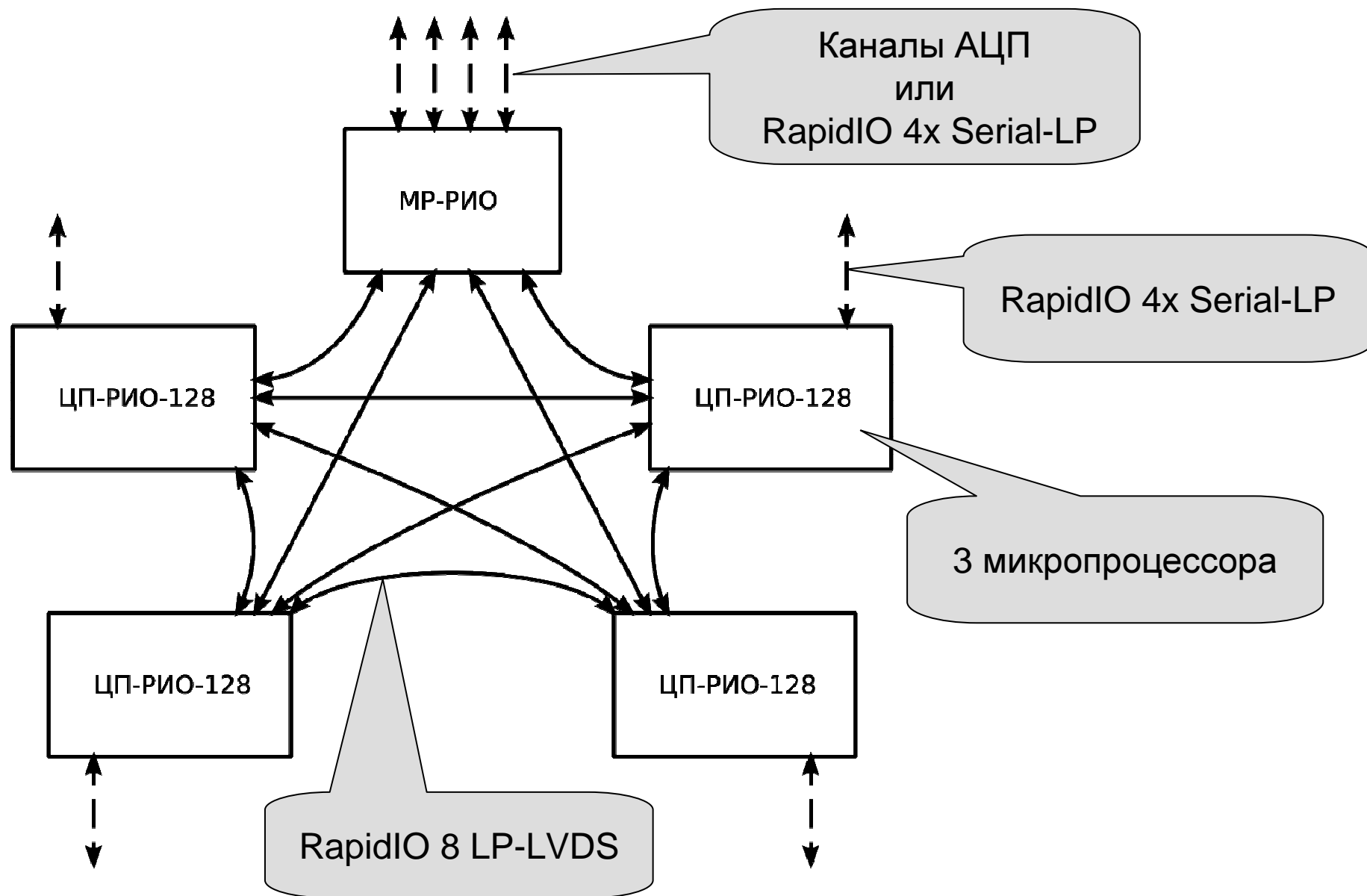
# Пример: обмены



# Пример: вычисления



# Базовая вычислительная ячейка



# Пример: результаты валидации

- Вручную было проведено размещение задач по процессорам и составлено XML-описание
- Объём описания: 88 КБ  
Число строк: 993
- Проведена проверка, что **предложенное размещение не перерасходует аппаратные ресурсы**
- Построена схема топологии коммуникационной среды