Особенности структурирования информационных систем

- Структурирование системы
- Моделирование управления
- Модульность
- Принцип информационной закрытости
- Связность модуля
- Сцепление модулей
- Метрики графа модулей

Проектирование

Проектирование

Структурирование системы

- Выделяем несколько подсистем
- Каждая подсистема решает определённый круг задач
- Определяем способ взаимодействия подсистем

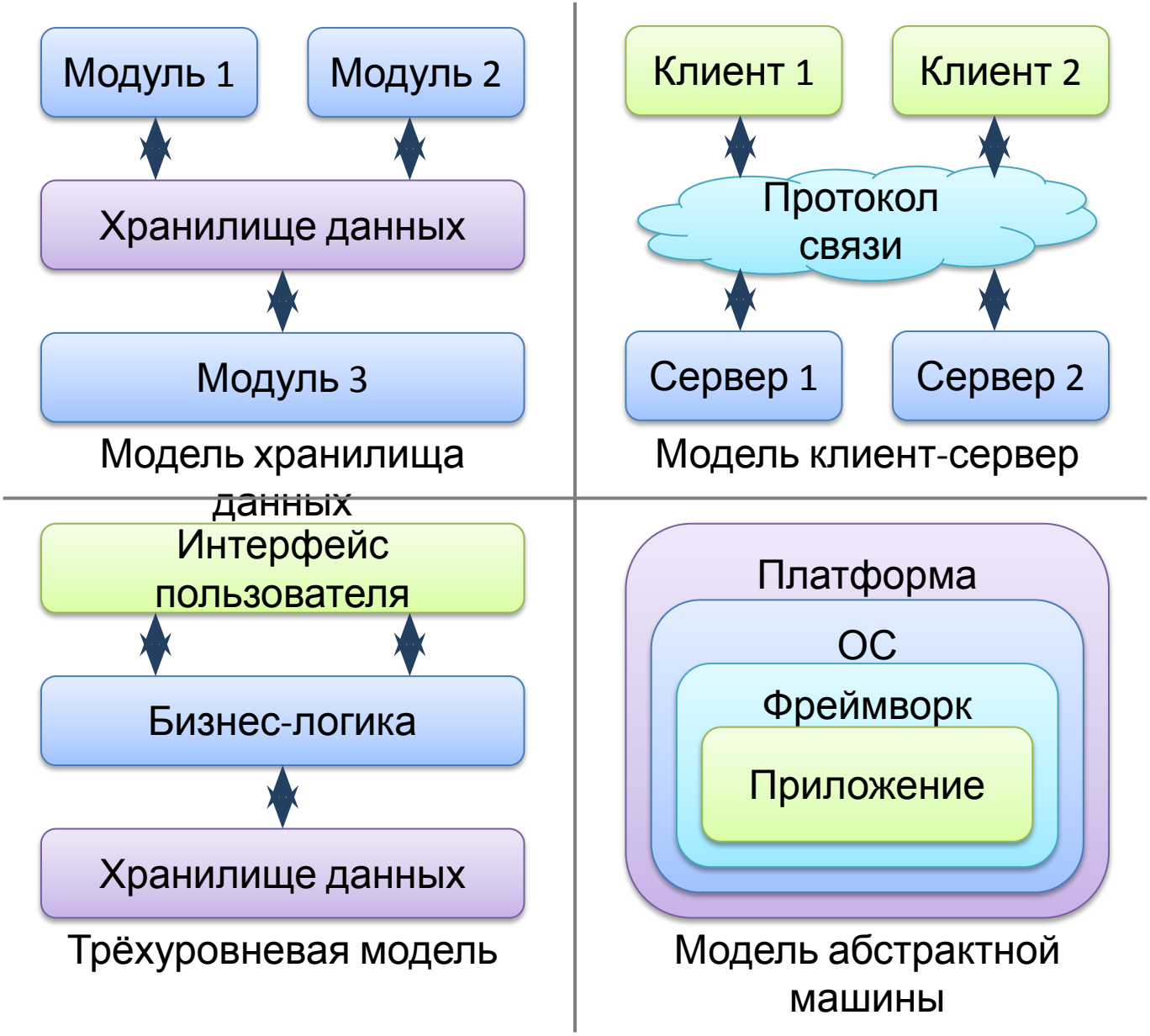
Моделирование управления

- Определяем управляющие связи между подсистемами
- «Кто кем командует?»

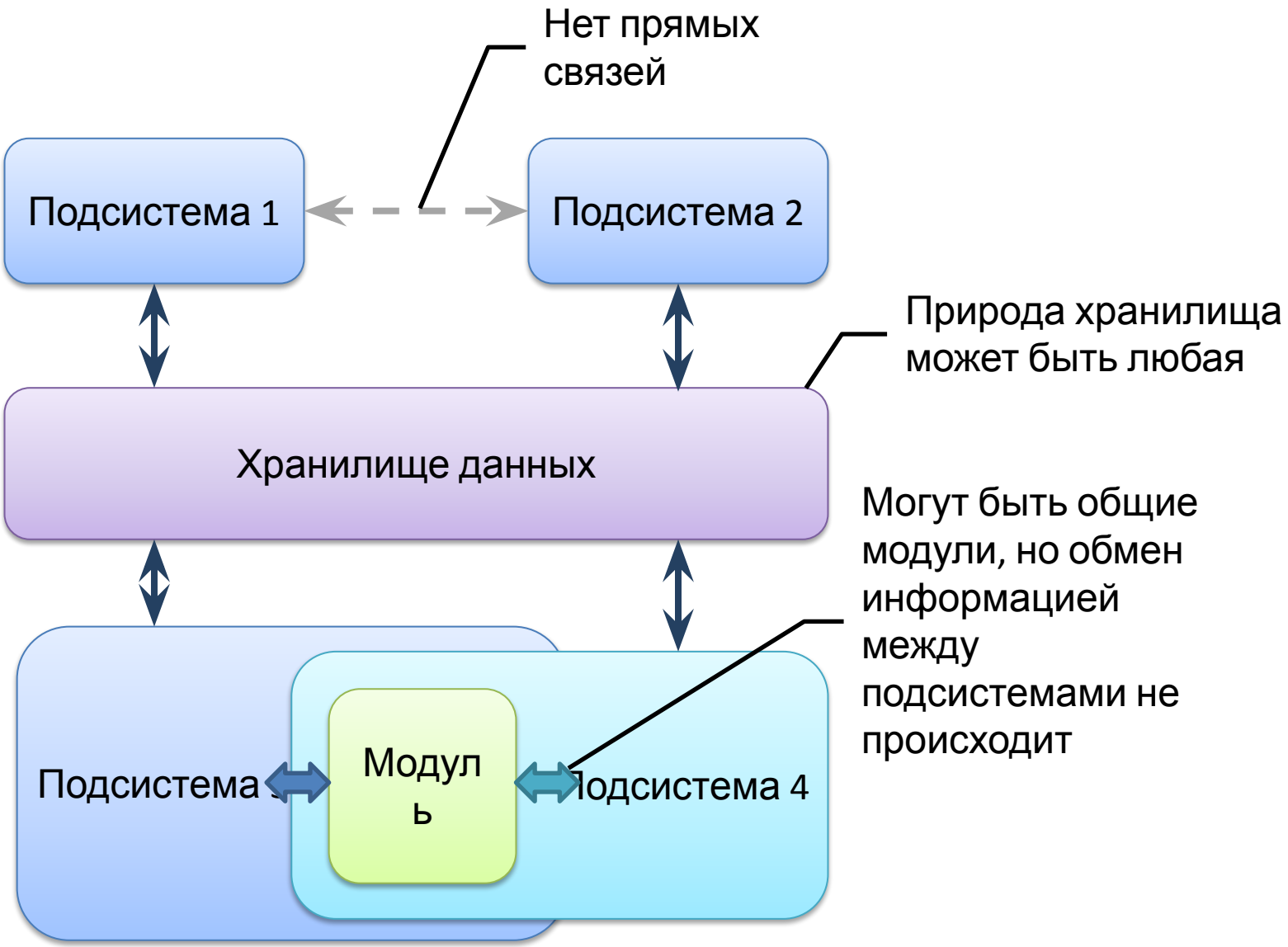
Декомпозиция подсистем

- Выделяем специализированные модули
- Каждый модуль решает узкий круг задач
- Разные подсистемы могут использовать один и тот же модуль

Структурирование системы



Модель хранилища данных



Модель хранилища данных

Недостатки

- Смена способа хранения данных требует переделки все системы
- Изменение структуры хранилища затронет все подсистемы, работающие с этими данными
- Низкая скорость взаимодействия между подсистемами
- Проблемы с реализацией одновременного доступа к хранилищу

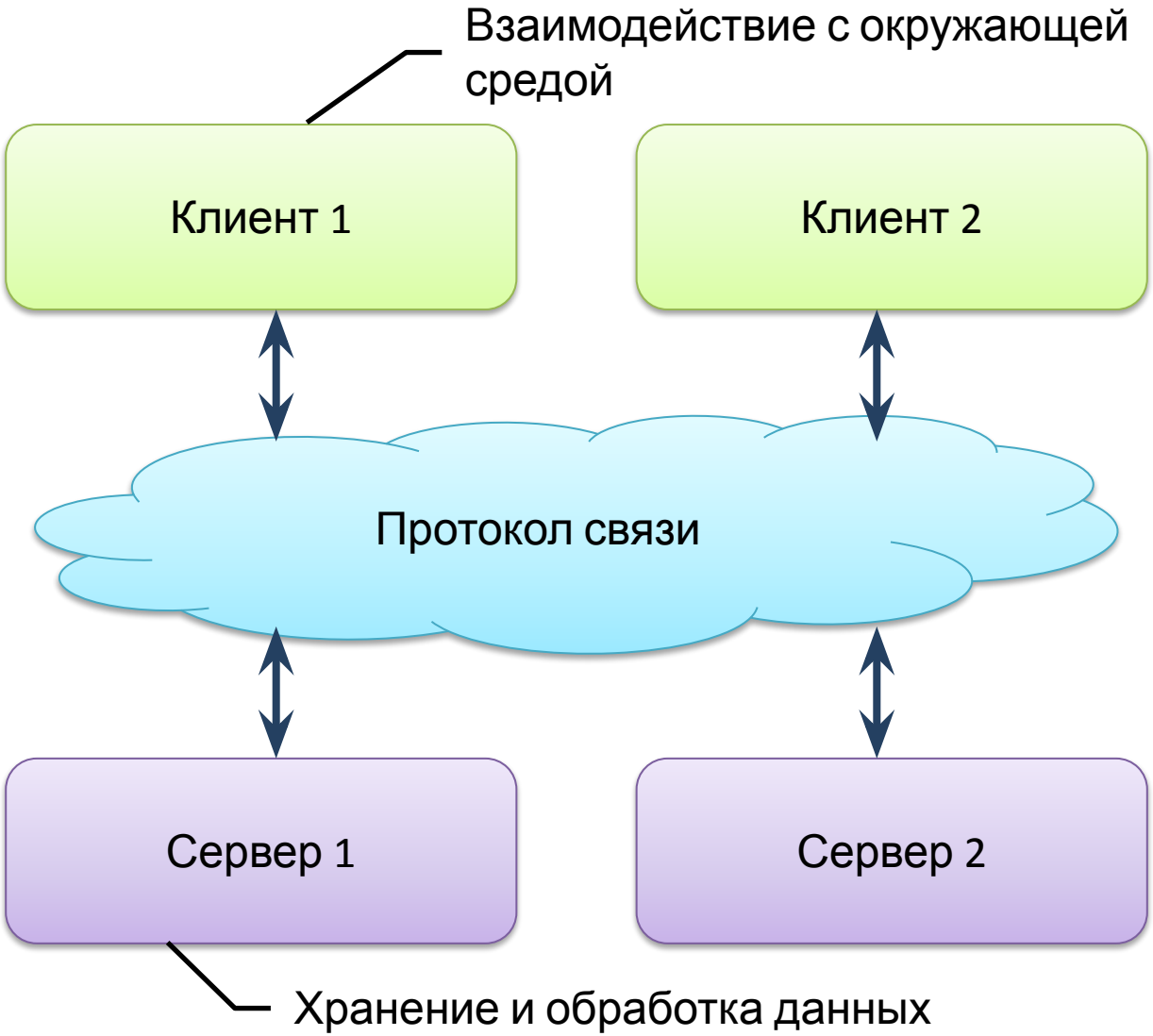
Преимущества

- Простота системы
- Легко расширять систему новыми подсистемами
- Можно использовать разные ЯП для подсистем
- Легко сохранять и восстанавливать состояние системы

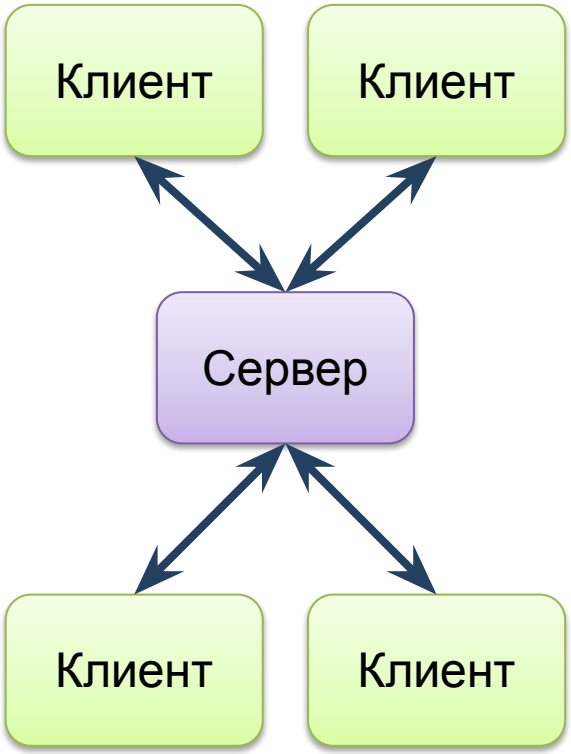
Используется

- Утилиты, работающие в пакетном режиме
- Веб-сайты

Модель клиент-сервер

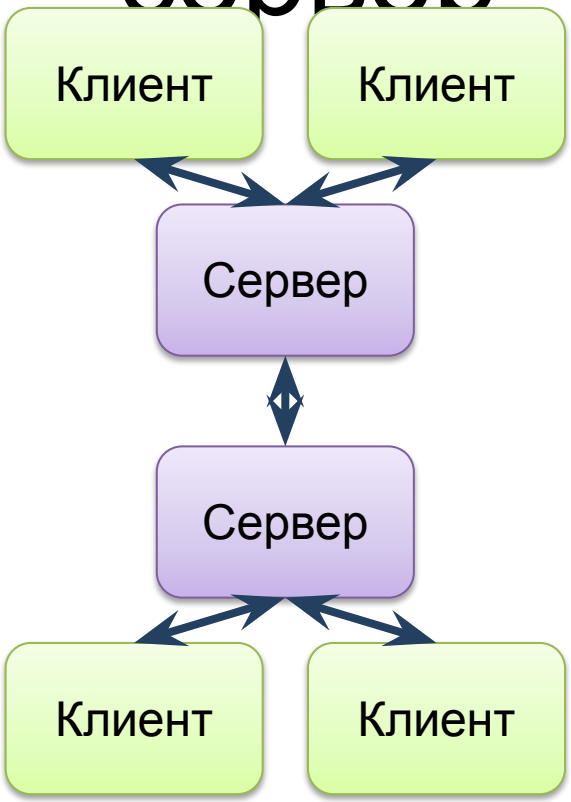


Варианты модели клиент-сервер



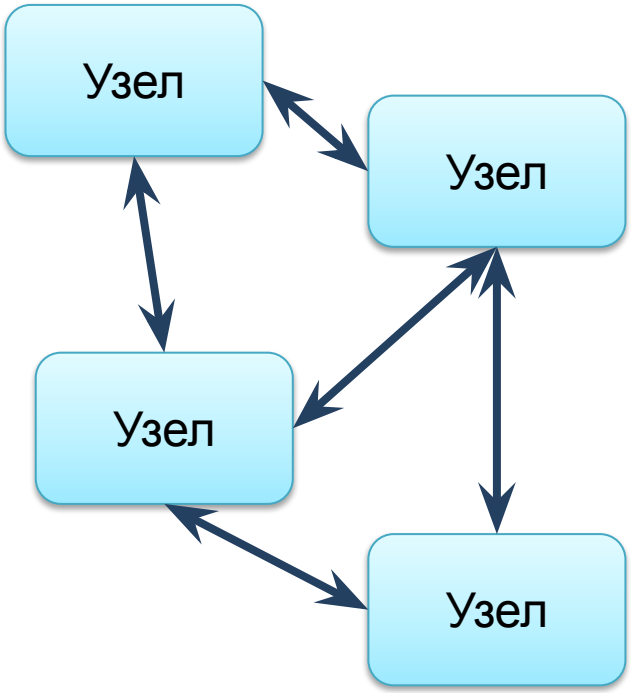
Централизованная

- Клиенты используют единый сервер
- Этот сервер является единой точкой отказа



Федеративная

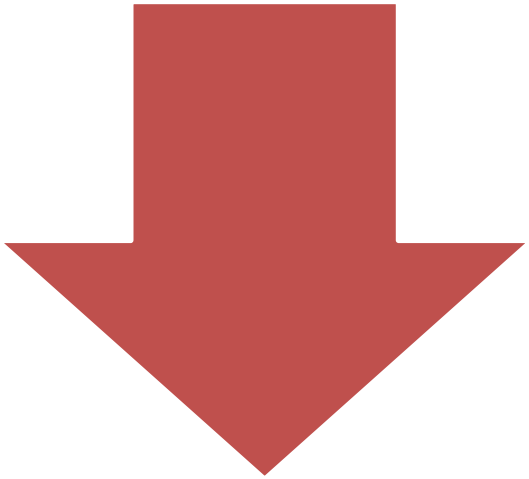
- Сервер обслуживает свою группу клиентов
- Серверы могут взаимодействовать между собой



Одноранговая

- Каждый узел может играть роль как клиента, так и сервера
- Проблемы с поиском соседних узлов

Централизация



Минусы

- Отказ сервера приводит к отказу системы
- Меньшая автономность клиентов
- Высоконагруженный сервер сложнее масштабировать, чем много дешёвых клиентов

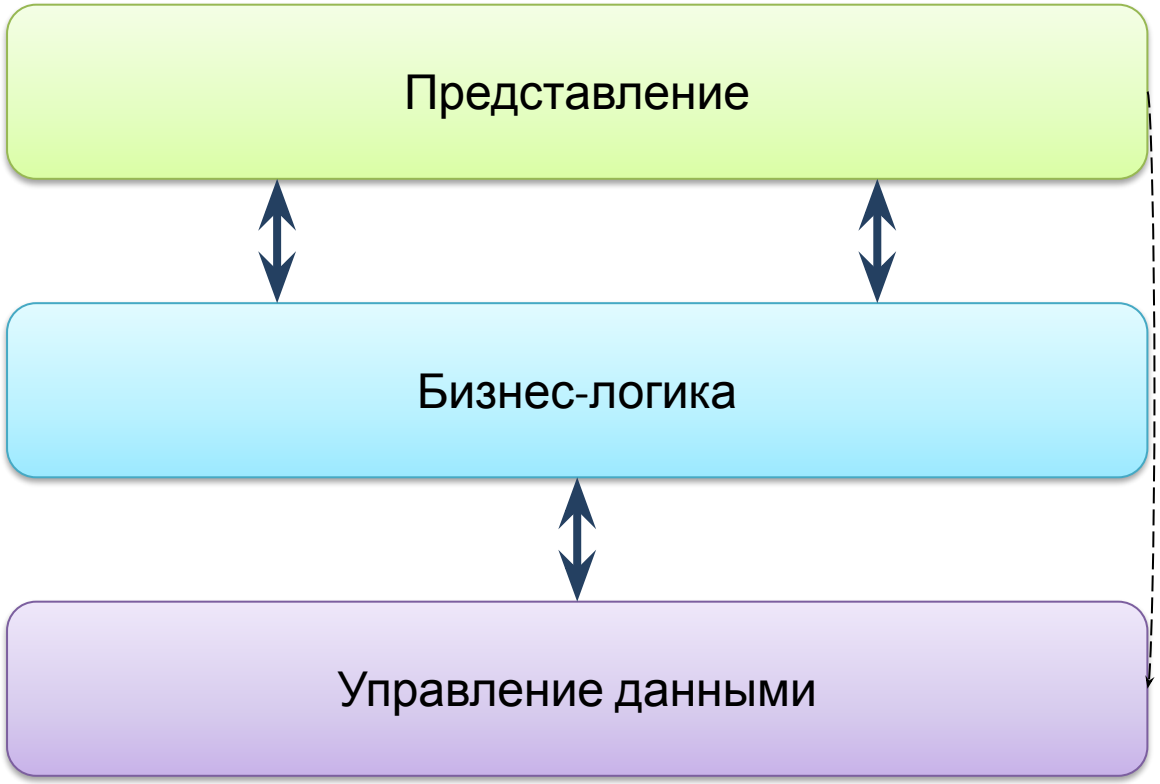
Плюсы

- Проще поддерживать данные актуальными
- Проще производить анализ работы системы
- Проще организовать резервирование и защиту данных
- Клиентские устройства проще заменять

Трёхуровневая модель

Сетевые приложения	Основная задача	Монолитные приложения
Слой представления		
Клиентское приложение	**Взаимодействие с пользователем**	Графический интерфейс
Слой бизнес-логики		
Сервер приложений	**Преобразование информации**	Модули логики
Слой управления данными		
Сервер баз данных	**Хранение состояния системы**	Хранилище в памяти Смешанные хранилища

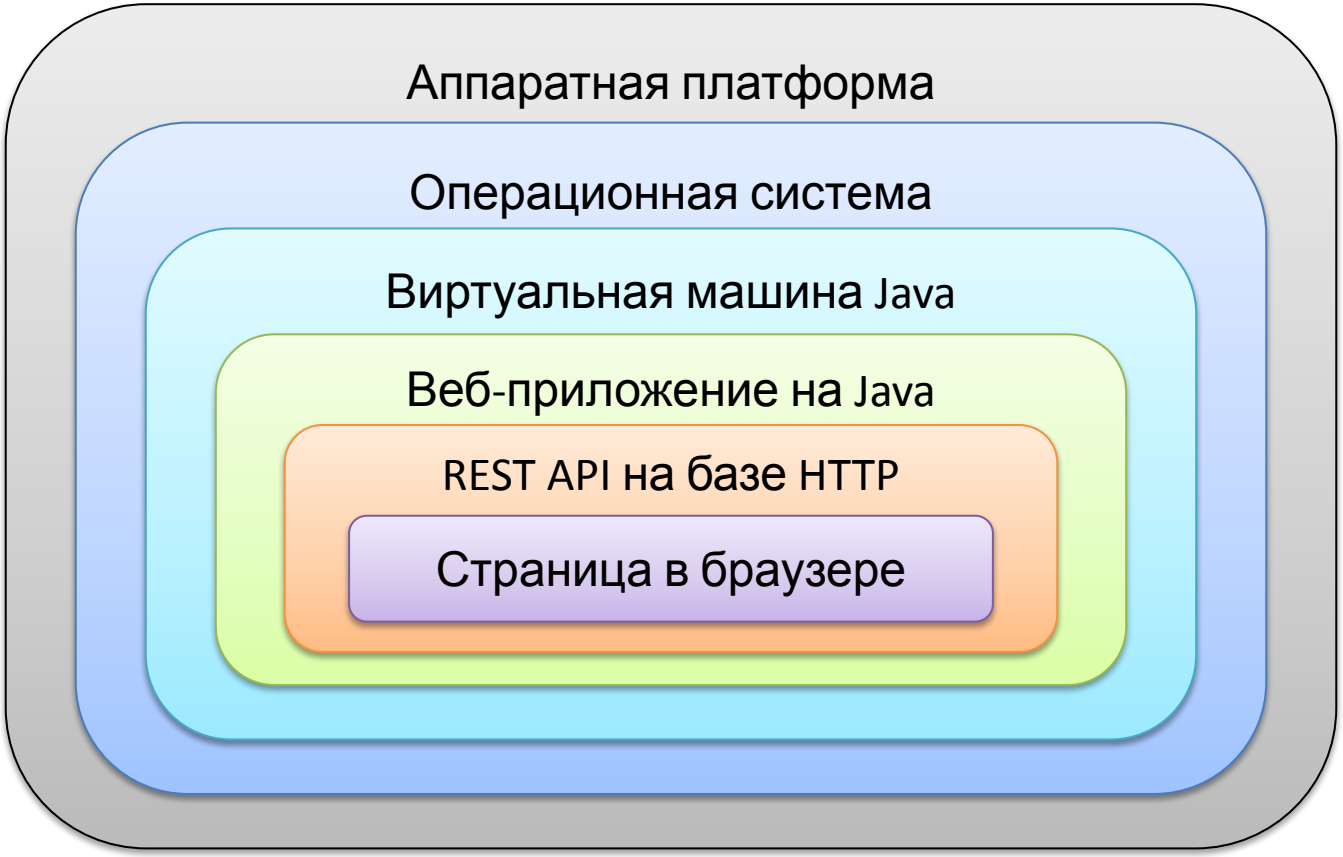
Трёхуровневая модель



У представления нет прямого доступа к хранилищу, как в MVC

Недостатки	Достоинства
• Логика вынуждена заниматься «переброской» данных туда-сюда.	• Проще модифицировать отдельные слои • Проще повторно использовать наработки в слоях представления и управления данными

Модель абстрактной машины



Недостатки

- Каждый слой абстракции увеличивает потребность в ресурсах
- Труднее решать задачи, которые задействуют несколько уровней

Достоинства

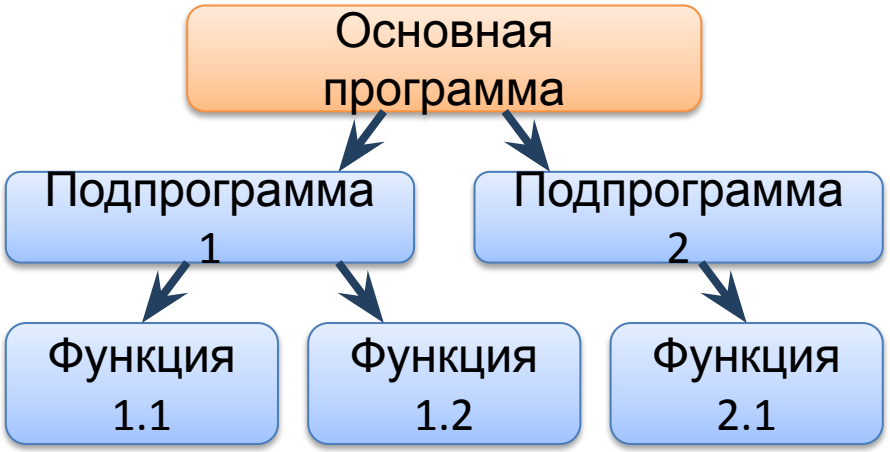
- Упрощается реализация каждого следующего слоя
- Возможна замена реализации одного из слоёв

Применение

- Фреймворки и библиотеки
- Приложения с возможностью скриптинга/автоматизации

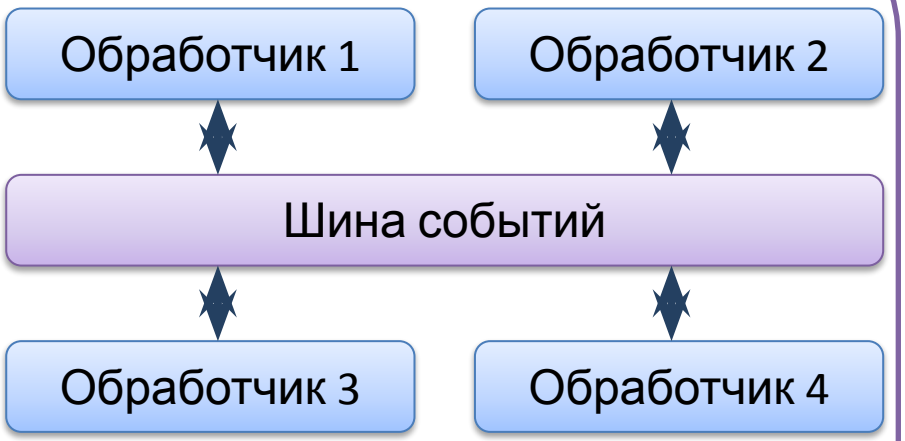
Моделирование управления

Централизованное управление

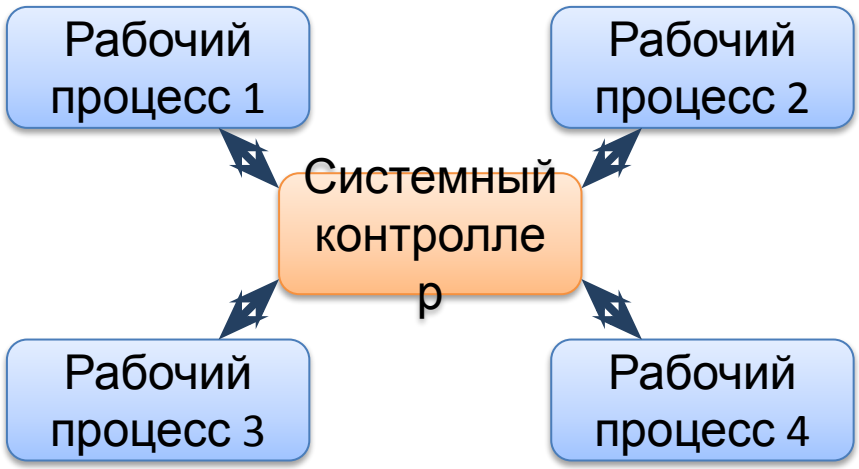


Модель «вызов-возврат»

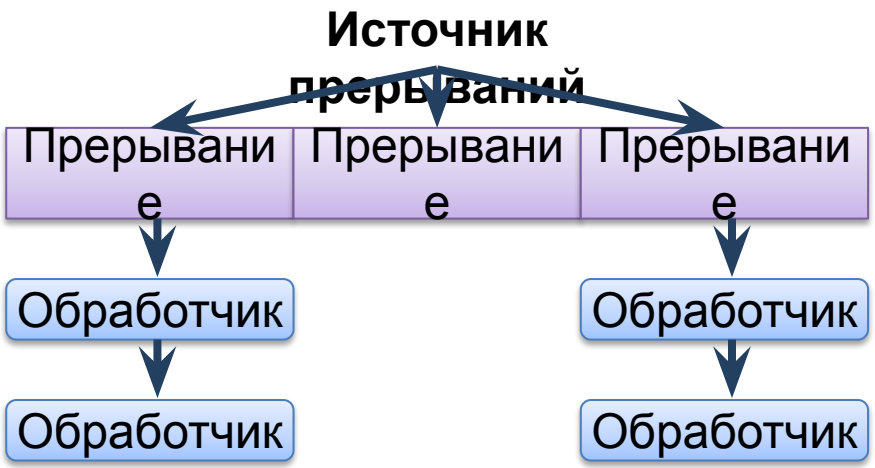
Событийное управление



Широковещательная модель

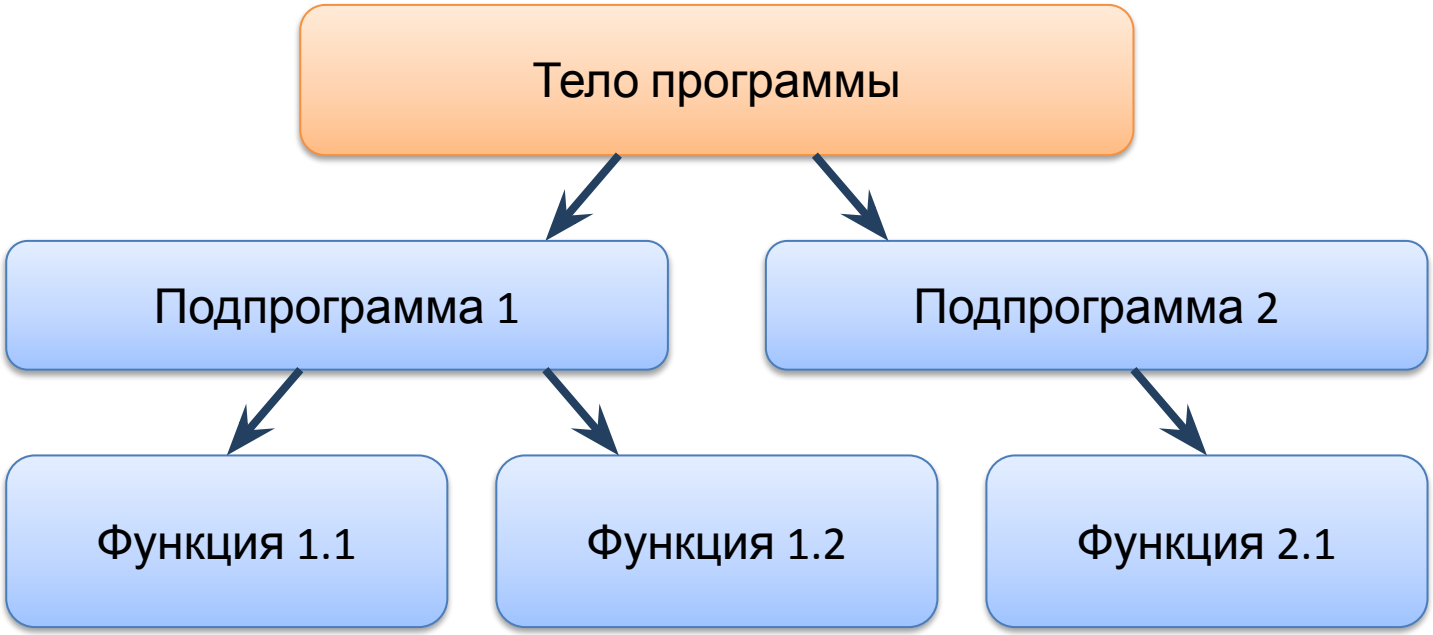


Модель менеджера



Модель, управляемая прерываниями

Модель «вызов-возврат»



Недостатки

- Не очень удобна, если требуется реагировать на поступающие события
- Плохо совместима с GUI

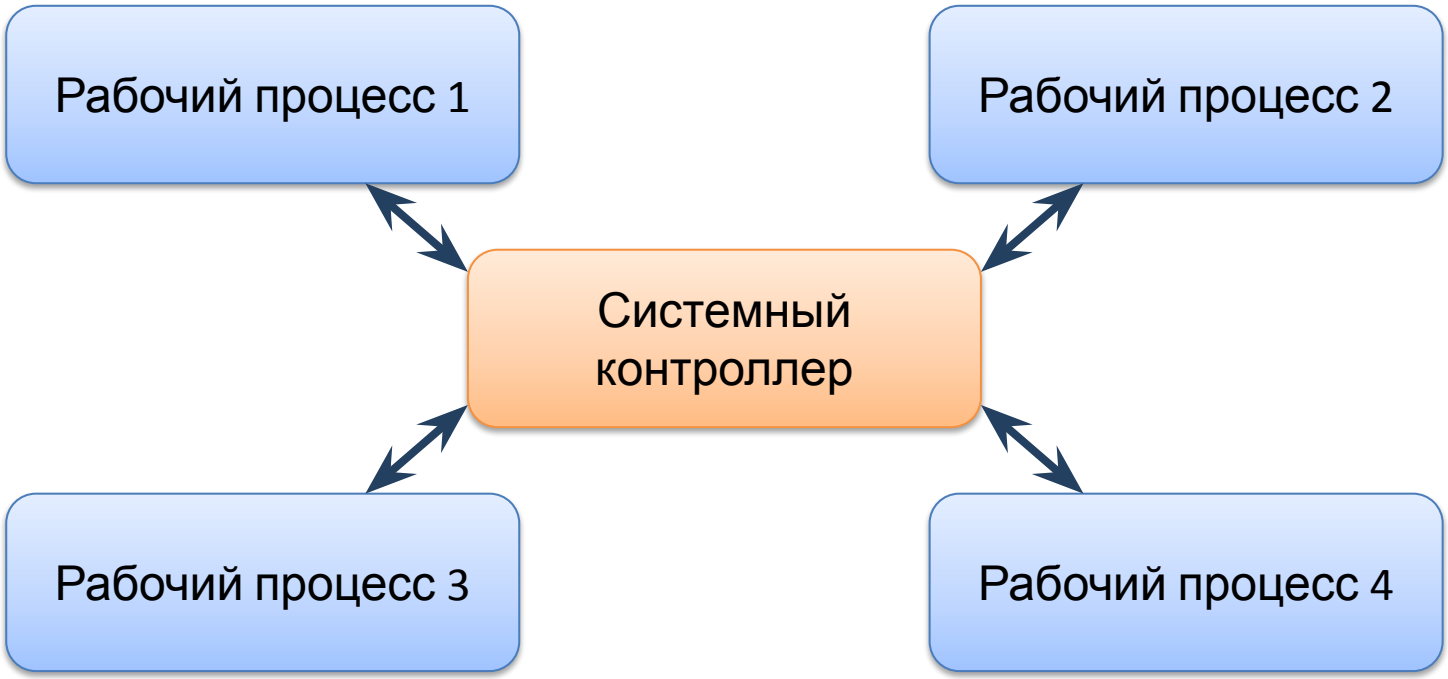
Достоинства

- Хорошо работает в пакетном режиме
- Очень проста

Применение

- Утилиты командной строки
- Некоторые сервисы (постоянно работающие программы)

Модель менеджера



Недостатки

- Имеет смысл только при параллельном выполнении нескольких процессов
- Проблемы «состояний гонки»

Достоинства

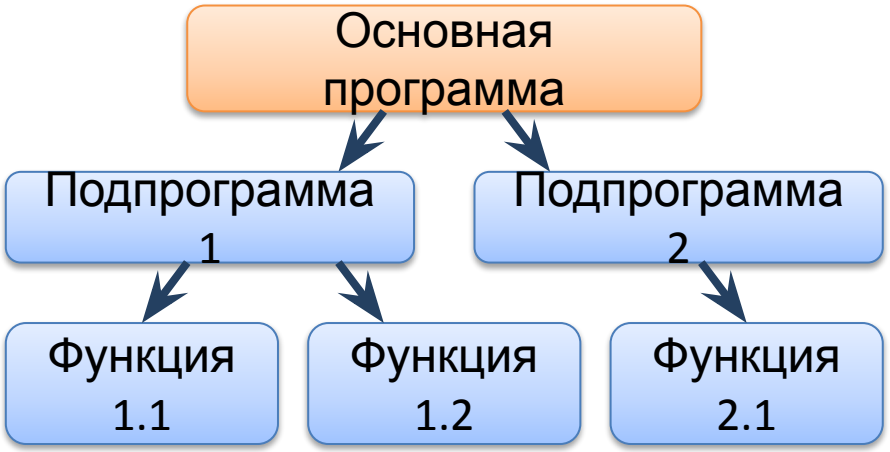
- Хорошо работает в приложениях массового обслуживания
- Сравнительно легко масштабируется

Применение

- Сетевые сервисы (серверная часть)
- Утилиты с графическим интерфейсом пользователя, выполняющие длительные действия

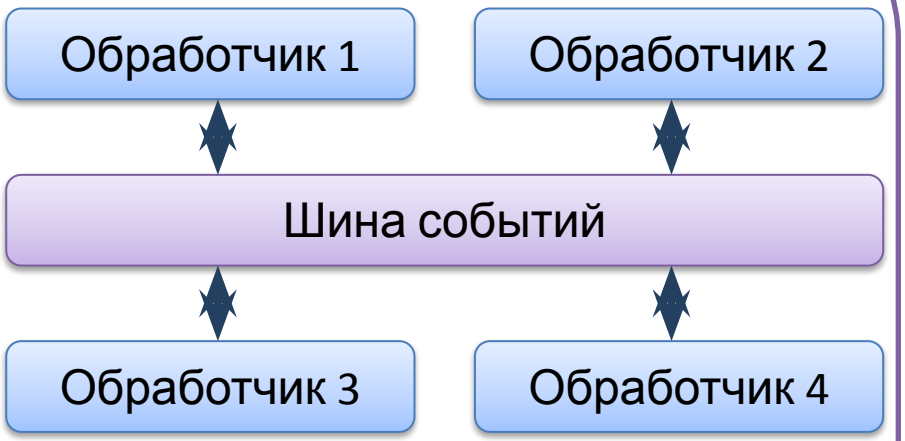
Моделирование управления

Централизованное управление

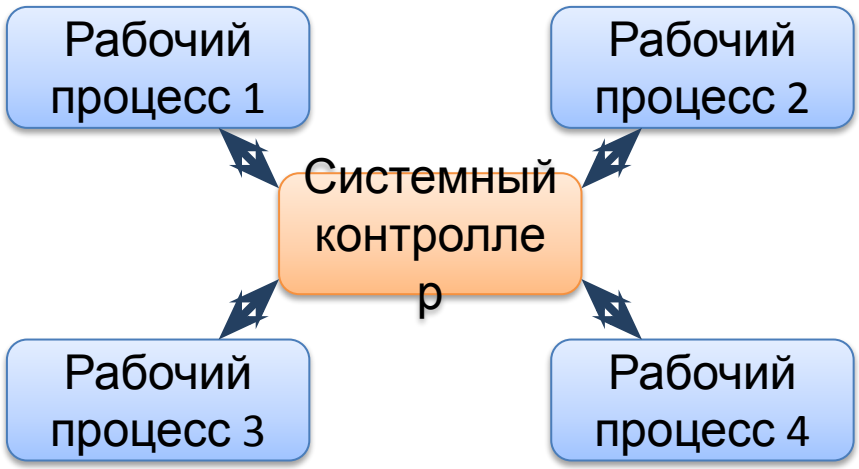


Модель «вызов-возврат»

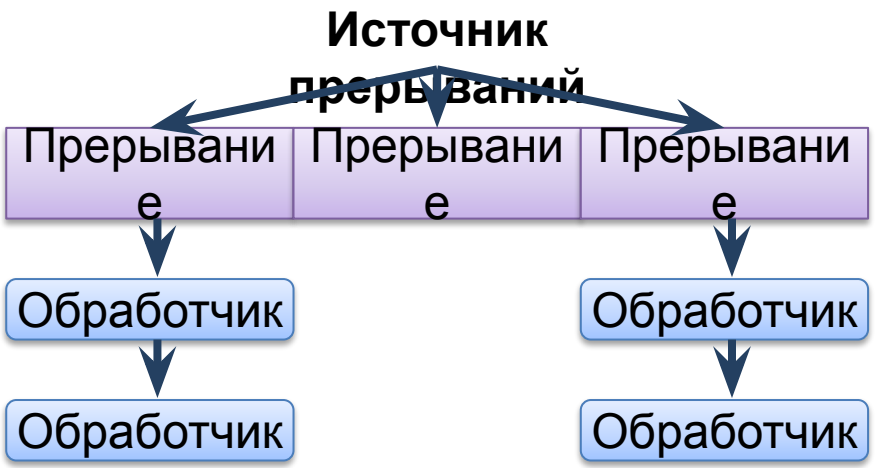
Событийное управление



Широковещательная модель

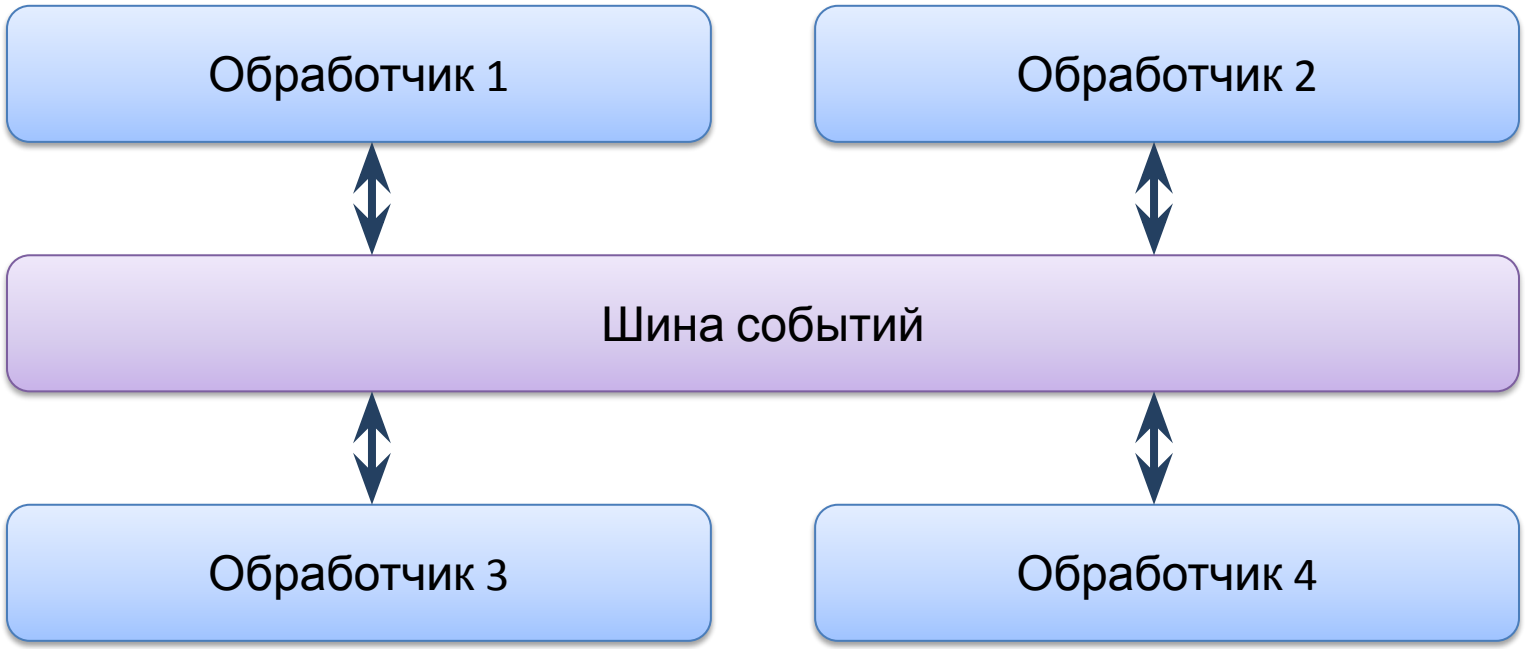


Модель менеджера



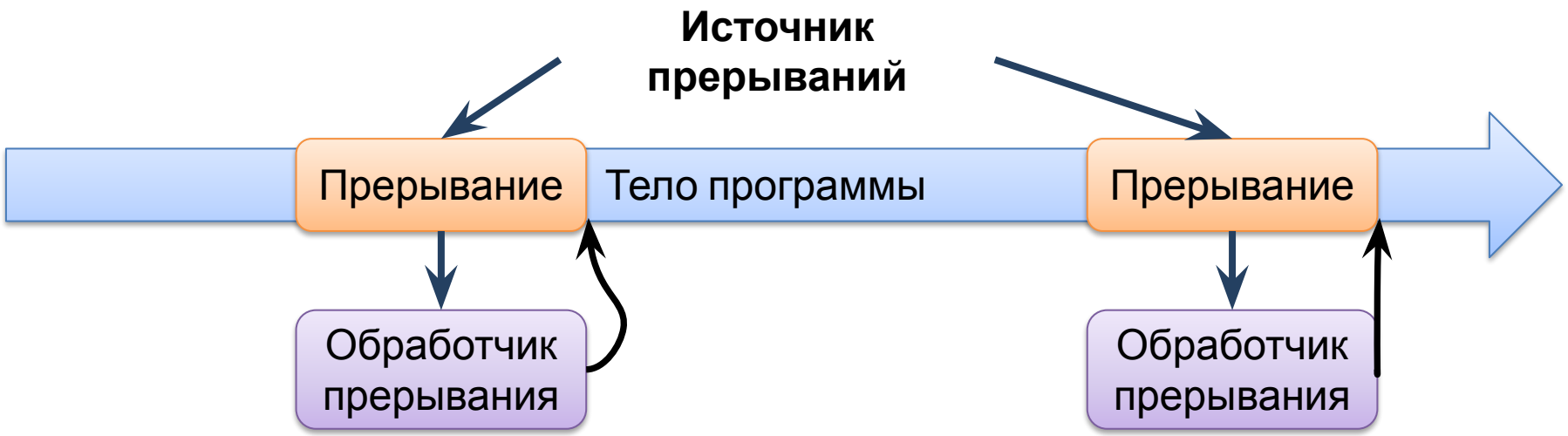
Модель, управляемая прерываниями

Широковещательная модель



Недостатки	Достоинства	Применение
• Сложные сценарии оказываются «размазаны» по множеству обработчиков событий	• Хорошо работает, когда нужно реагировать на заранее неизвестную последовательность входных событий • Сравнительно легко расширяется	• Утилиты с графическим интерфейсом пользователя • Асинхронные приложения, ориентированные на ввод/вывод

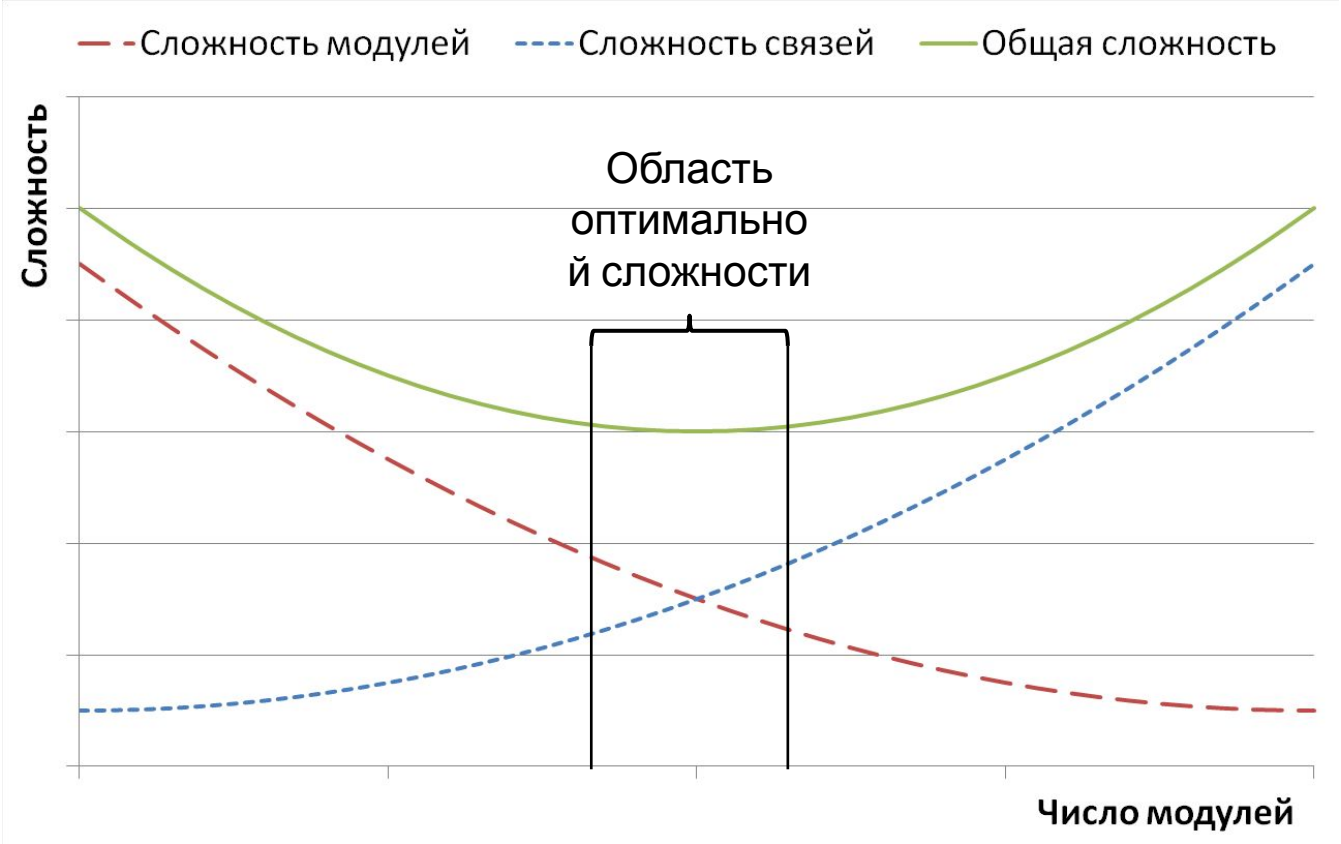
Модель, управляемая прерываниями



Недостатки	Достоинства	Применение
• Возможные конфликты одновременно пришедших прерываний • Обработчик прерывания должен восстановить контекст выполнения по окончанию своей работы	• Позволяет разделить программу на низкоприоритетный «основной цикл» и обработчики важных событий	• Микроконтроллеры • Встраиваемые системы

Модульность

Модульность — свойство системы, которая может подвергаться декомпозиции на ряд внутренне связанных и слабо зависящих друг от друга модулей.



Хороший модуль проще использовать, чем переписать заново.

Принцип информационной закрытости

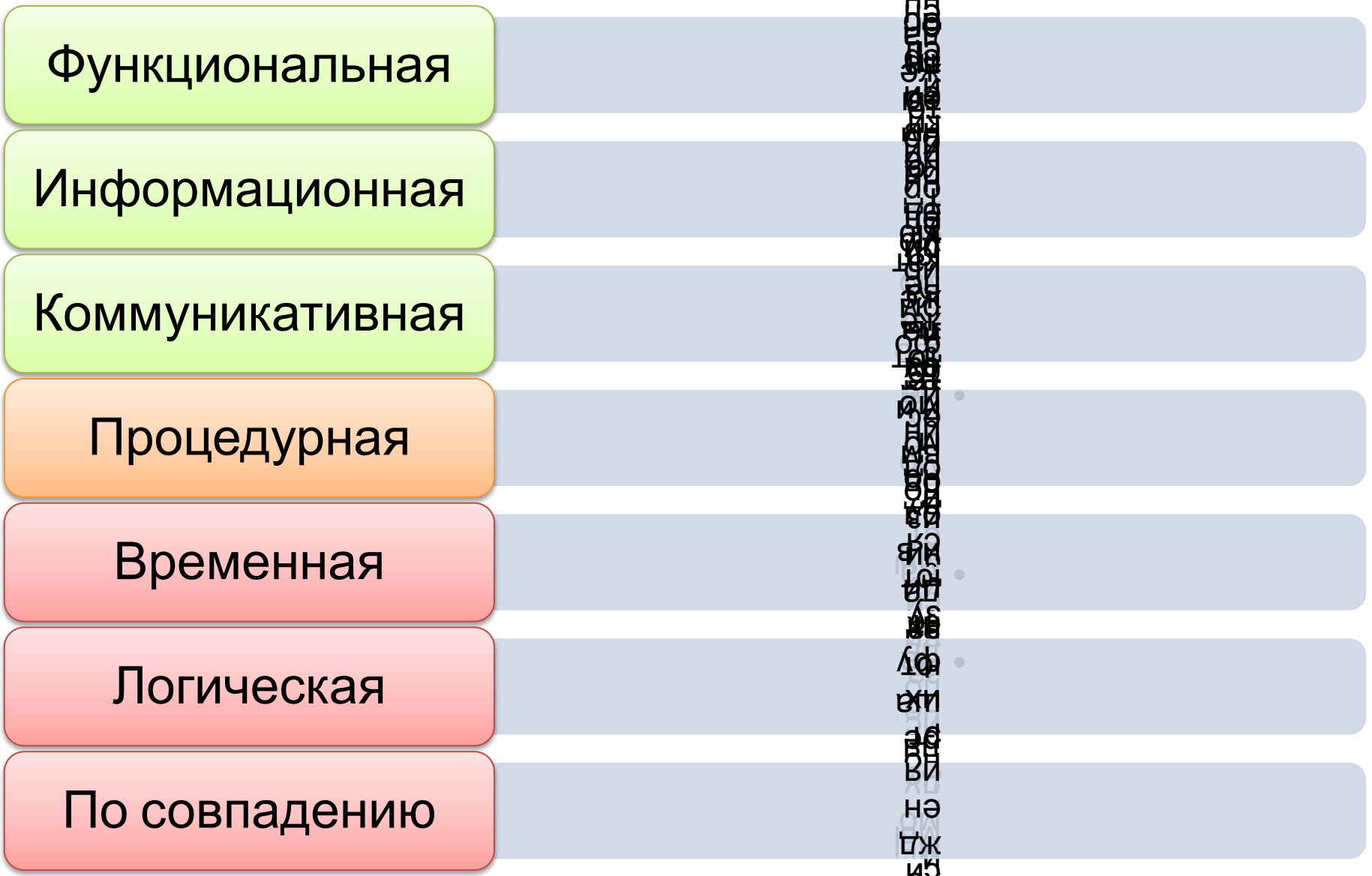
Сущность

- Модули по возможности независимы друг от друга
- Доступ к содержимому модуля ограничен
- Модули обмениваются только той информацией,

Достоинства

- Параллельная разработка модулей системы разными коллективами разработчиков
- Упрощается модификация системы, реже сталкиваемся с «каскадом изменений»

Связность модуля



Алгоритм определения связности

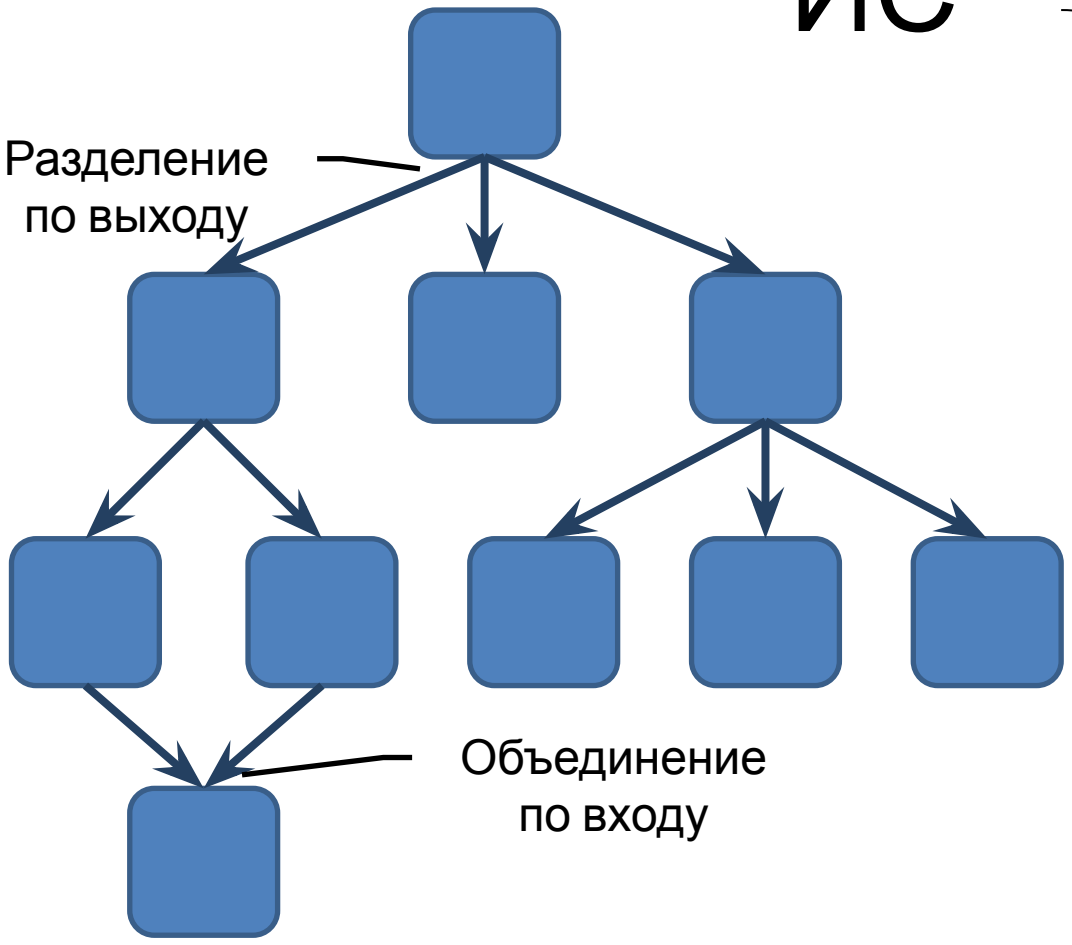
Правило параллельной цепи: если все действия модуля имеют несколько уровней связности, то модулю присваивают самый сильный из них.

Правило последовательной цепи: если действия в модуле имеют разные уровни связности, то модулю присваивают самый слабый из них.

Сцепление модулей

- По данным
- По образцу
- По управлению
- По внешним ссылкам
- По общей области
- По содержанию

Метрики сложности структуры ИС



Число связей в полном графе

$$E_C = \frac{1}{2}N(N-1)$$

Число связей в графе-дереве

$$E_T = N - 1$$

Невязка графа

$$Nev = \frac{E - E_T}{E_C - E_T} =$$

$$= \frac{2(E - N + 1)}{(N-1)(N-2)}$$