

Программная инженерия

Анализ и проектирование
программного обеспечения – 3

Вопросы

- Архитектура:
 - Как сделать: Алгоритм построения архитектуры
 - Какая: Образцы проектирования, выбор шаблона архитектуры
- Описание архитектуры с помощью UML
 - Activity diagram
 - Sequence diagram
 - Class diagram

Как сделать?

Какая?

ПОСТРОЕНИЕ АРХИТЕКТУРЫ

Алгоритм построения архитектуры – 1/2

- Разбейте систему на замкнутые модули
- Сравните со стандартными архитектурами
→ улучшите декомпозицию
– Создайте несколько (sic!)
- Сделайте выбор среди альтернативных
- Добавьте к классам, полученным из требований, классы, обеспечивающие согласование с выбранной архитектурой
- Примените шаблон проектирования, если найдете полезный

Алгоритм построения архитектуры – 2/2

- Распределите классы по пакетам
 - Практическое правила:
 - 4-8 пакетов на уровне
 - В больших проектах использовать иерархию
- Проверить признаки «хорошего» разбиения:
 - Связность – высокая
 - Сцепление – низкое
- Рассмотрите возможность добавления facade-класса (объекта) для управления интерфейсами пакетов

Образцы проектирования – 1/2

- Образец проектирования – найденная опытным путем комбинация компонентов, решающая общие задачи проектирования
- Полезные образцы:
 - Facade – единый интерфейс для множества объектов различных классов
 - State – объект ведет себя в соответствии с текущим состоянием
 - Iterator – позволяет выбирать пути «посещения» объектов в коллекции во время выполнения клиентского кода

Образцы проектирования – 2/2

- Observer – обеспечивает реакцию нужных элементов на изменения в источнике данных
- Interpreter – интерпретирует выражения на языке формальной грамматики

Выбор архитектуры – 1/3

Признак

- Есть поток данных между обрабатывающими станциями?
- Обрабатывающие станции ожидают получения входных данных, чтобы начать работу?
- Процессы выполняются параллельно?

Архитектура

- Архитектура последовательных пакетов
- Архитектура каналов и фильтров
- Архитектура параллельных взаимодействующих процессов

Выбор архитектуры – 2/3

Признак

- Процесс обеспечивает обслуживание пользовательских процессов?
- Процесс реагирует только на происходящие события?
- Приложение состоит из процессов, которые выполняются по сценарию?

Архитектура

- Архитектура клиент-сервер
- Системы, управляемые событиями
- Образец проектирования «интерпретатор»

Выбор архитектуры – 3/3

Признак

- Приложение строится для хранения данных?
- Существует упорядочение по уровням?

Архитектура

- Репозиторные архитектуры
- Уровневые архитектуры

Activity diagram

Sequence diagram

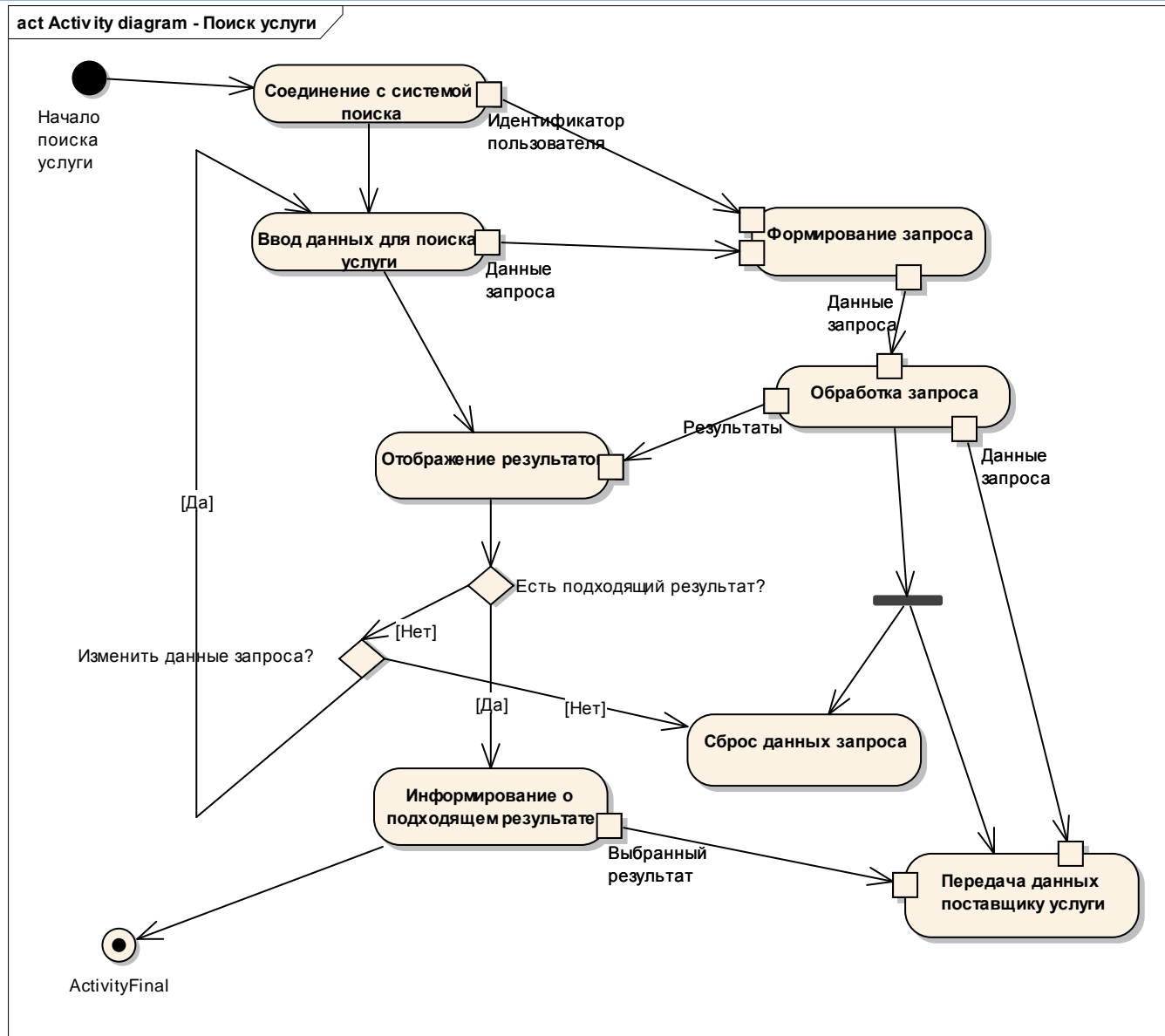
Class diagram

ДИАГРАММЫ UML

Архитектура последовательных пакетов – 1/2

- Лучше всего описывается с помощью DFD (Data Flow Diagram)
- Ближе всего в UML к этой диаграмме: Activity Diagram

Архитектура последовательных пакетов – 2/2



Сравнение DFD и AD

Data Flow Diagram

- Описывает потоки данных
- Легко привязывается к объектам (экземплярам классов), которые передаются
- Подразумевают одновременность обработки данных

Activity Diagram

- Описывают одновременно потоки управления и потоки данных
- Позволяют описать одновременность, но описание тяготеет к описанию последовательной обработки

Нотация: AD



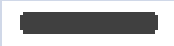


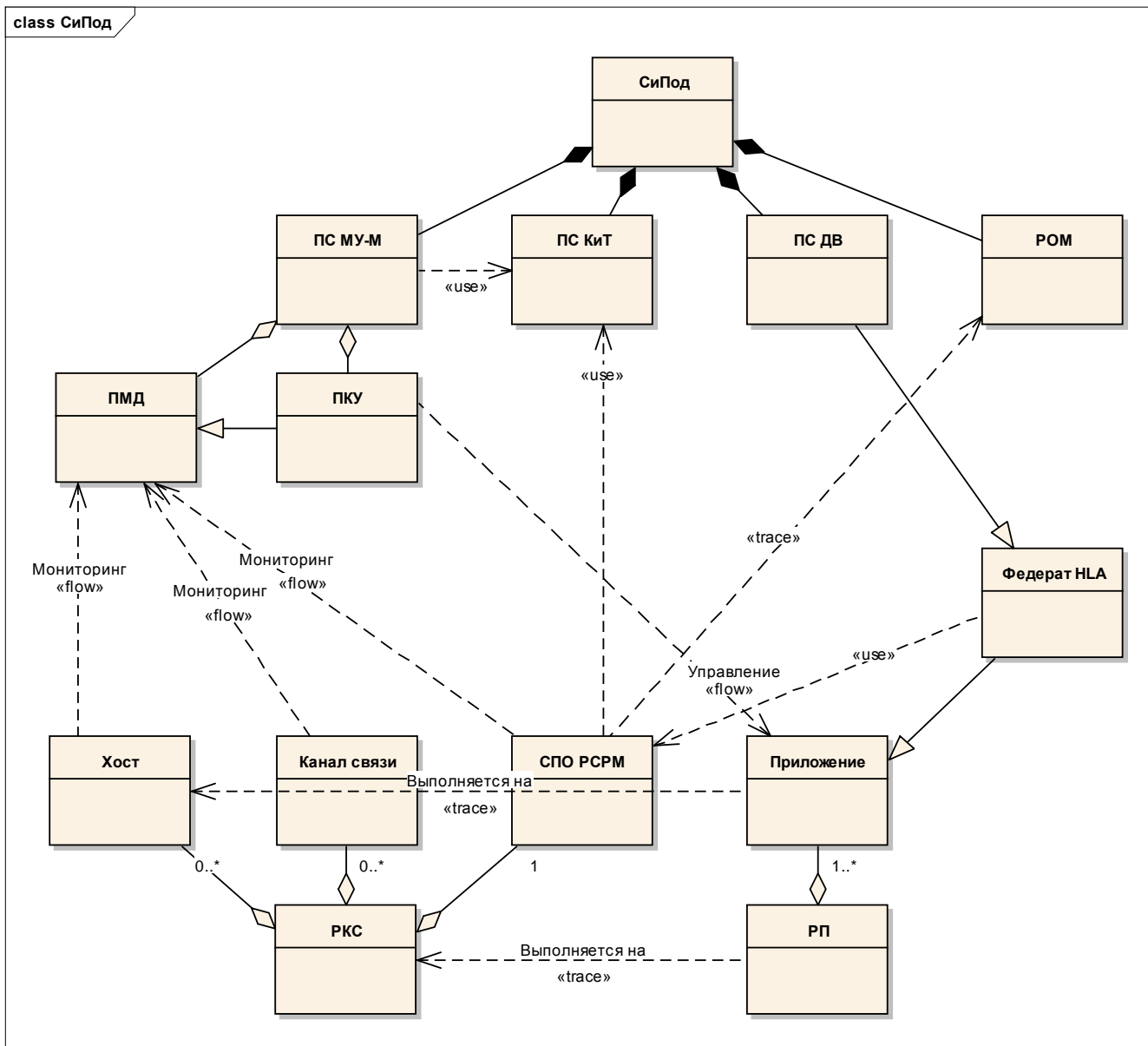
Блок	Описание
	Действие (action)
	Принятие решения (decision) Условие выбора дальнейшего пути (может делить на 2 и более путей)
	Начало / конец параллельного выполнения
	Начало обработки
	Конец обработки


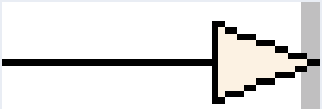
Диаграмма классов – 1/2

- Наиболее общий вид диаграммы, отображающий взаимосвязь классов в системе
- Может использоваться на любом этапе и уровне описания системы
- Представляет собой набор прямоугольников, соединенных различными связями

Диаграмма классов – 2/2



Нотация: диаграмма классов – 1/3

Связь	Описание
	<p>Ассоциация (association)</p> <ul style="list-style-type: none">• Реализуется в ЯП как указатель (ссылка)• Позволяет «перемещаться» от одного объекта к другому• Может связывать разные классы и класс сам с собой
	<p>Обобщение (generalization, inheritance)</p> <ul style="list-style-type: none">• От более специального класса (наследника) к более общему

Нотация: диаграмма классов – 2/3

Связь	Описание
	Реализация (realization, implementation)
	Агрегация (aggregation) <ul style="list-style-type: none">• Связь part-of (от части к целому)
	Композиция (composition) <ul style="list-style-type: none">• Связь part-of (от части к целому)• Предполагает, что целое состоит только из указанных компонентов
	Зависимость (dependency)

Нотация: Диаграмма классов – 3/3

- Мощность отношения (кратность) определяет количество экземпляров связанных такой связью. Может быть:
 - 0..1 – Ноль или один экземпляр
 - Количество жен
 - 1 – Строго один экземпляр
 - У человека только одна мать
 - 0..* или * – Ноль или более экземпляров
 - Количество мест работы
 - 1..* – Один или более экземпляров
 - У кошки есть хотя бы одно место, где она спит

Диаграмма последовательности – 1/2

- Показывает взаимодействие объектов (обмен между ними сигналами и сообщениями), упорядоченное по времени, с отражением продолжительности обработки и последовательности их проявления
- Обычно уточняет описание

Диаграмма последовательности – 2/2

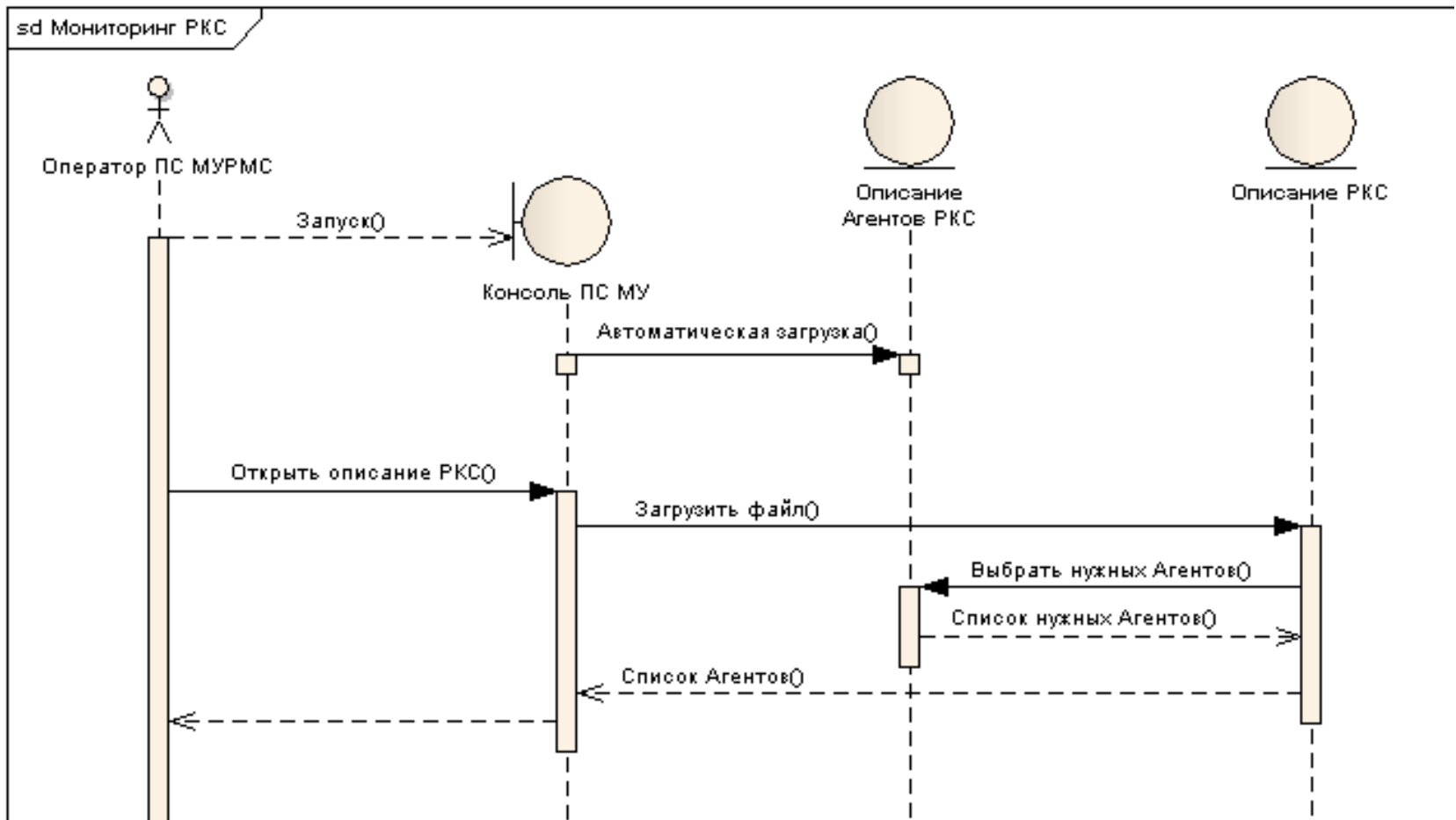


Диаграмма развертывания – 1/3

- Описывает физическое развертывание артефактов [программных компонентов] на узлах
- Типы узлов:
 - Узел устройства (Device Node)
 - Физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения (хосты)
 - Узел среды выполнения (Execution Environment Node)
 - Программный вычислительный ресурс, который: а) выполняется на узле б) предоставляет собой сервис, выполняющий другие исполняемые программные элементы

Диаграмма развертывания – 2/3

dd Deployment of Components

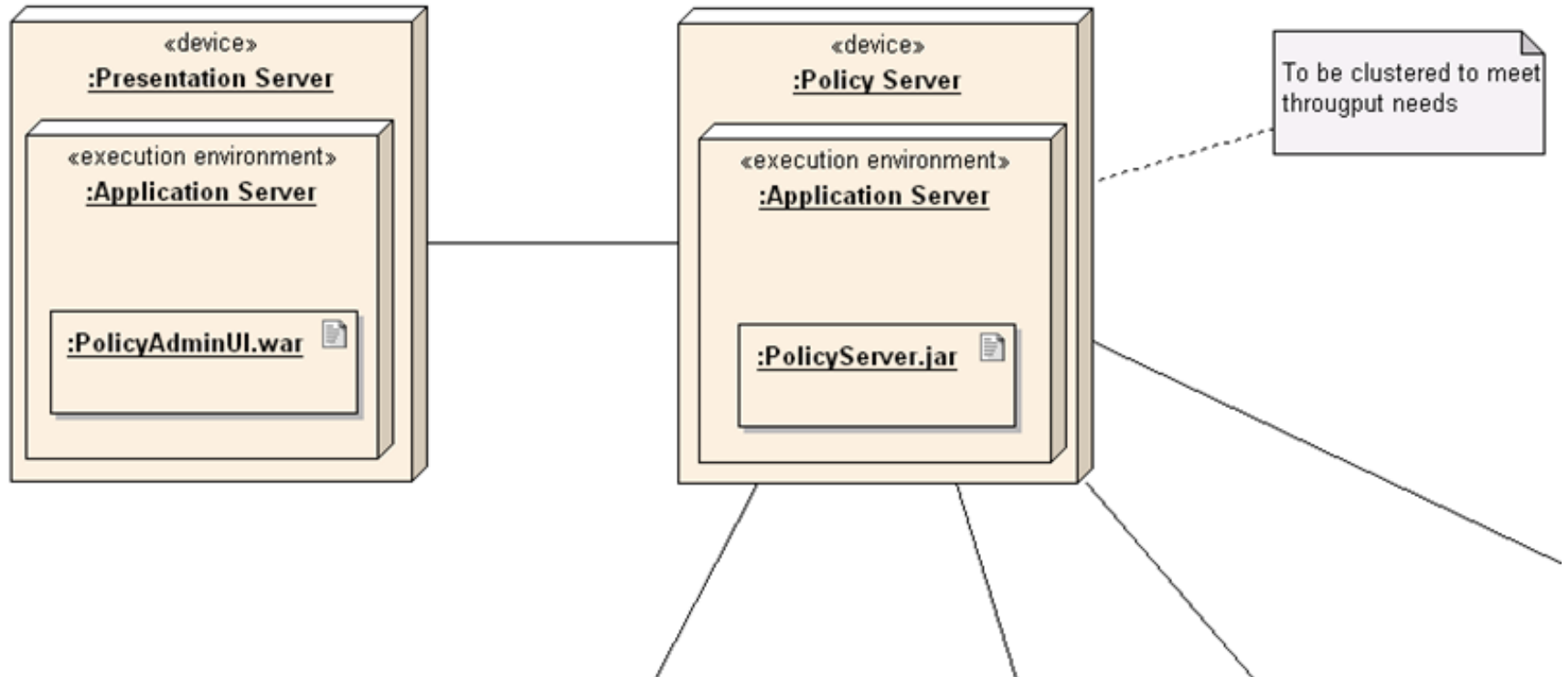
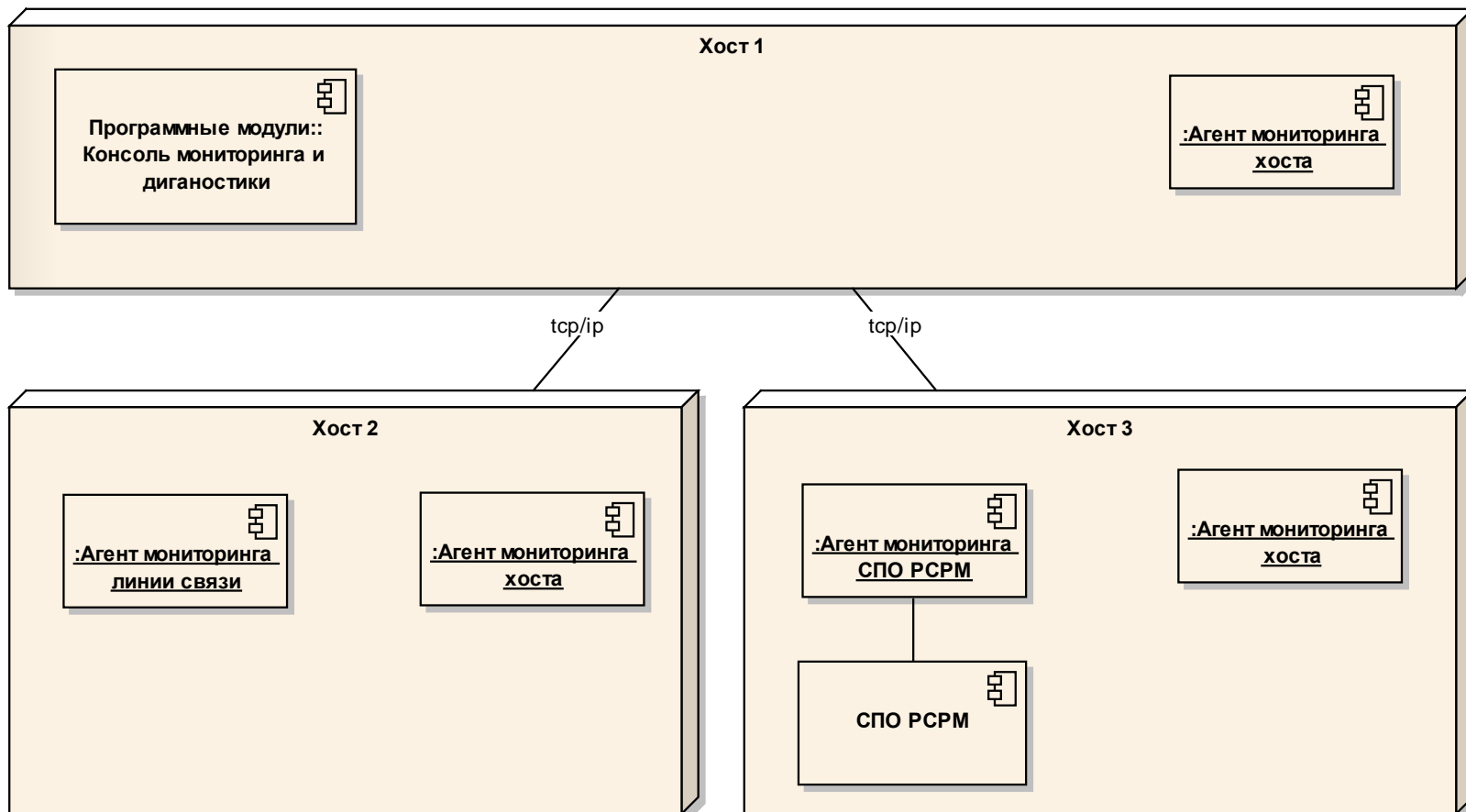


Диаграмма развертывания – 3/3

deployment Размещение программных модулей



Все пропало

Ужас-ужас

ПРАКТИЧЕСКИЕ ЗАДАНИЯ