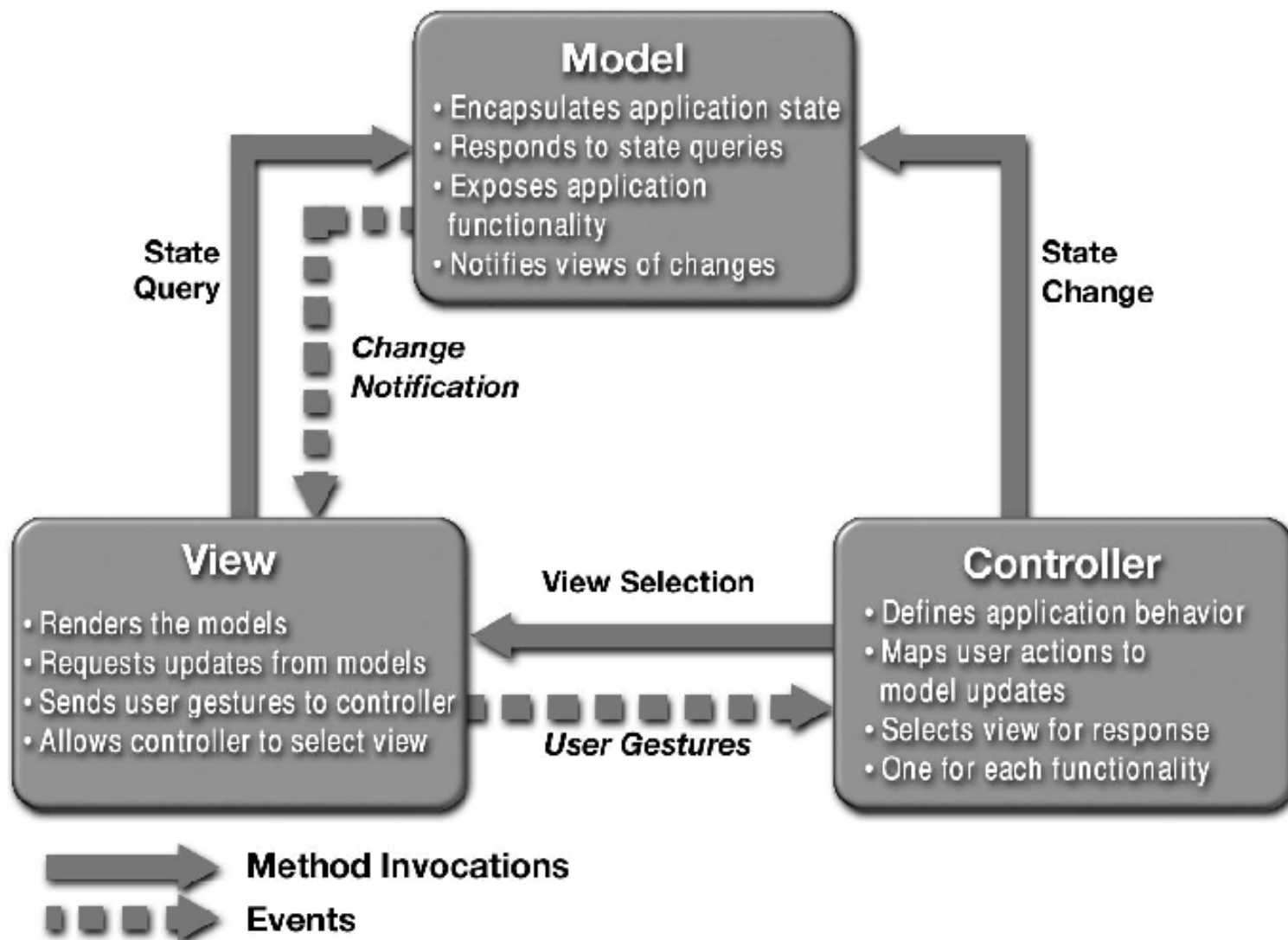


Архитектура Model-View-Controller и веб-приложения

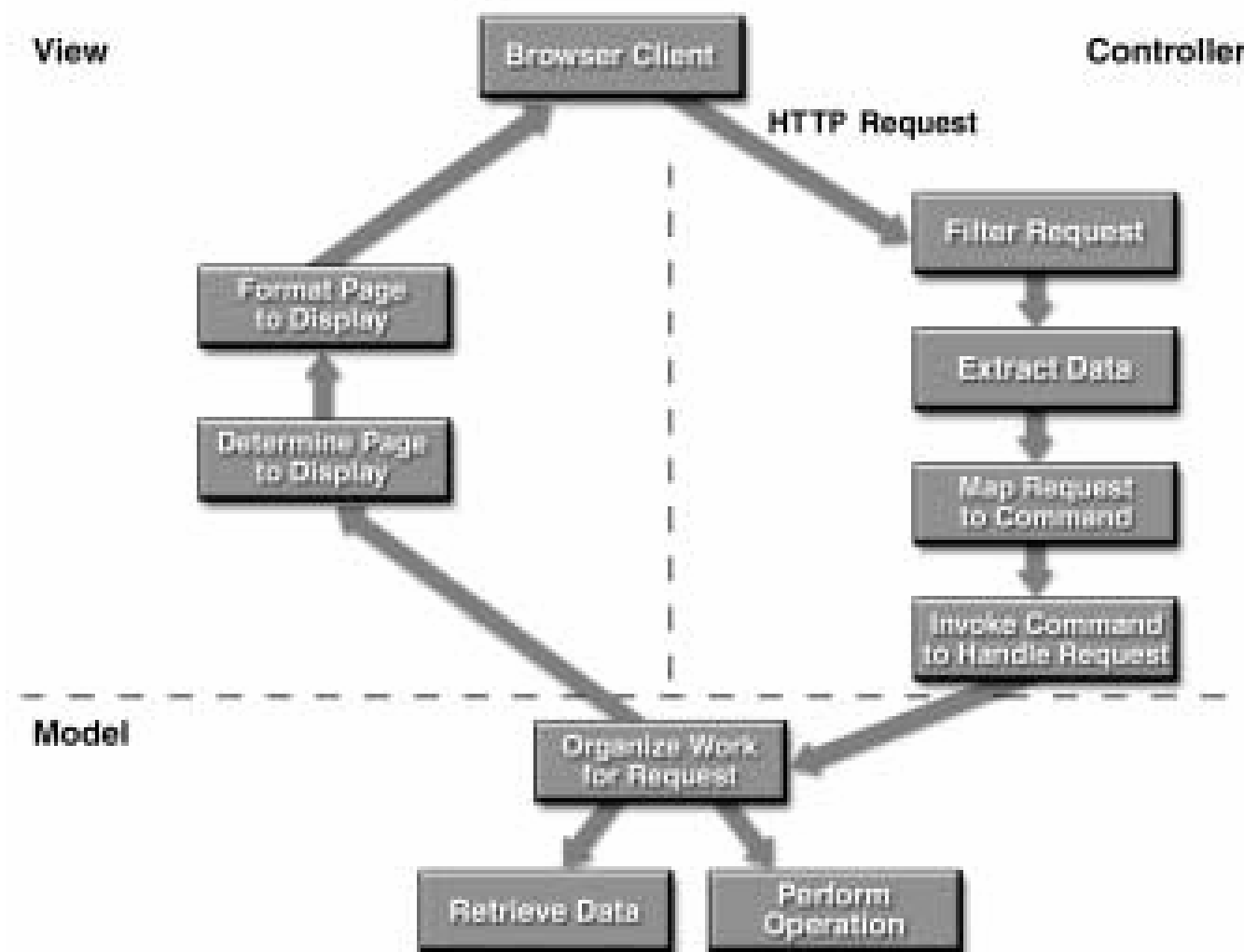
Архитектура (шаблон) MVC



Достоинства архитектуры MVC

- Уменьшение воздействия изменений
- Упрощение сопровождения (в том числе нахождения и исправления ошибок)
- Независимость от типа клиента и повторное использование кода
- Специализация труда разработчиков
- Централизация управления навигацией, безопасностью, журналированием и другими инфраструктурными функциями приложения

Архитектура MVC для веб-приложения



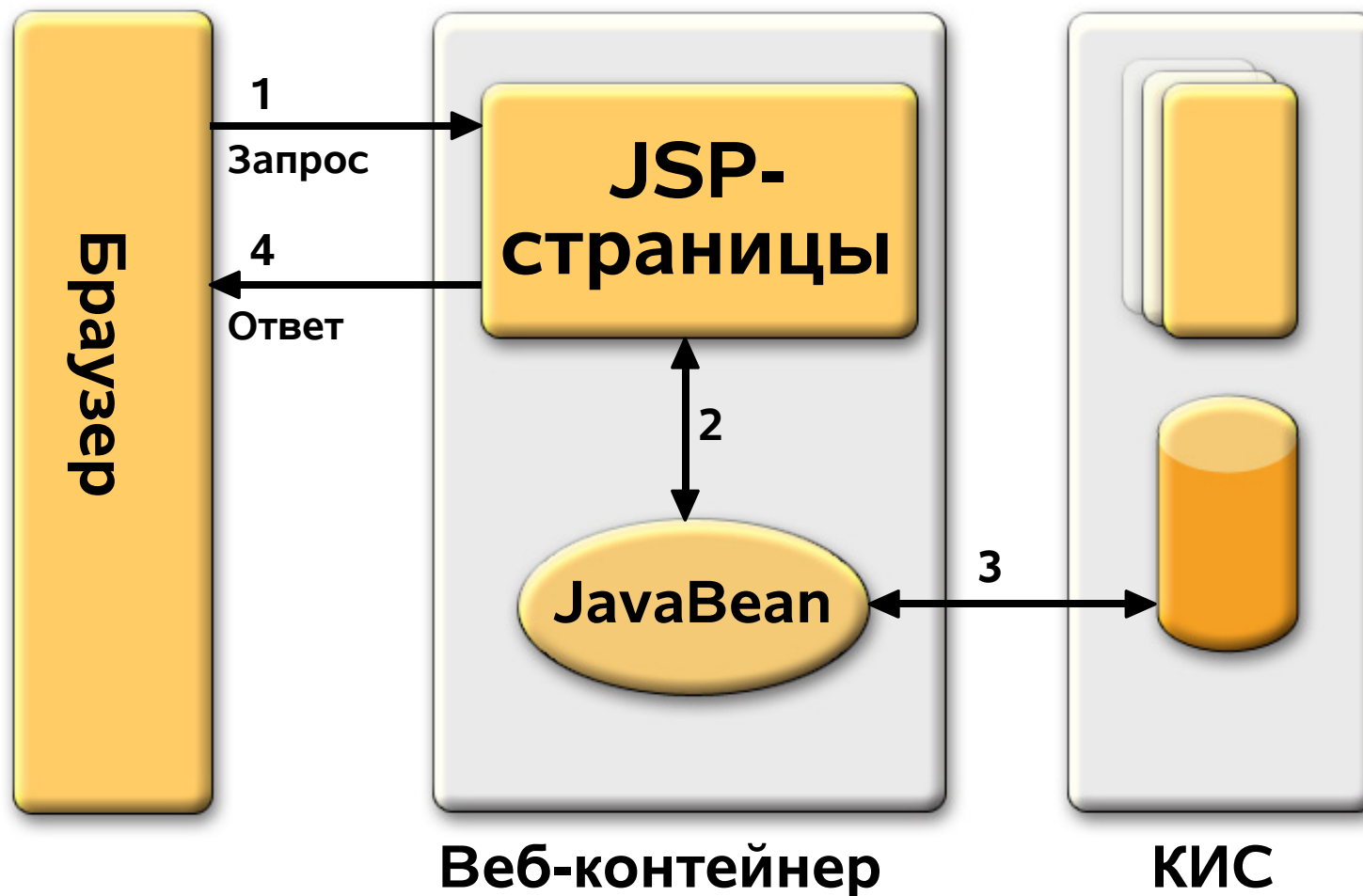
Типовые архитектуры веб-приложений на платформе Java EE



Эволюция архитектуры MVC

1. Без MVC
2. MVC Model 1 (на базе JSP-страниц)
3. MVC Model 2 (на базе сервлета)
4. Каркасы веб-приложений
 - > Struts, Tapestry, ...
5. Стандартный каркас веб-приложений
 - > JavaServer Faces (JSR-127)

Архитектура Model 1 (на базе JSP-страниц)

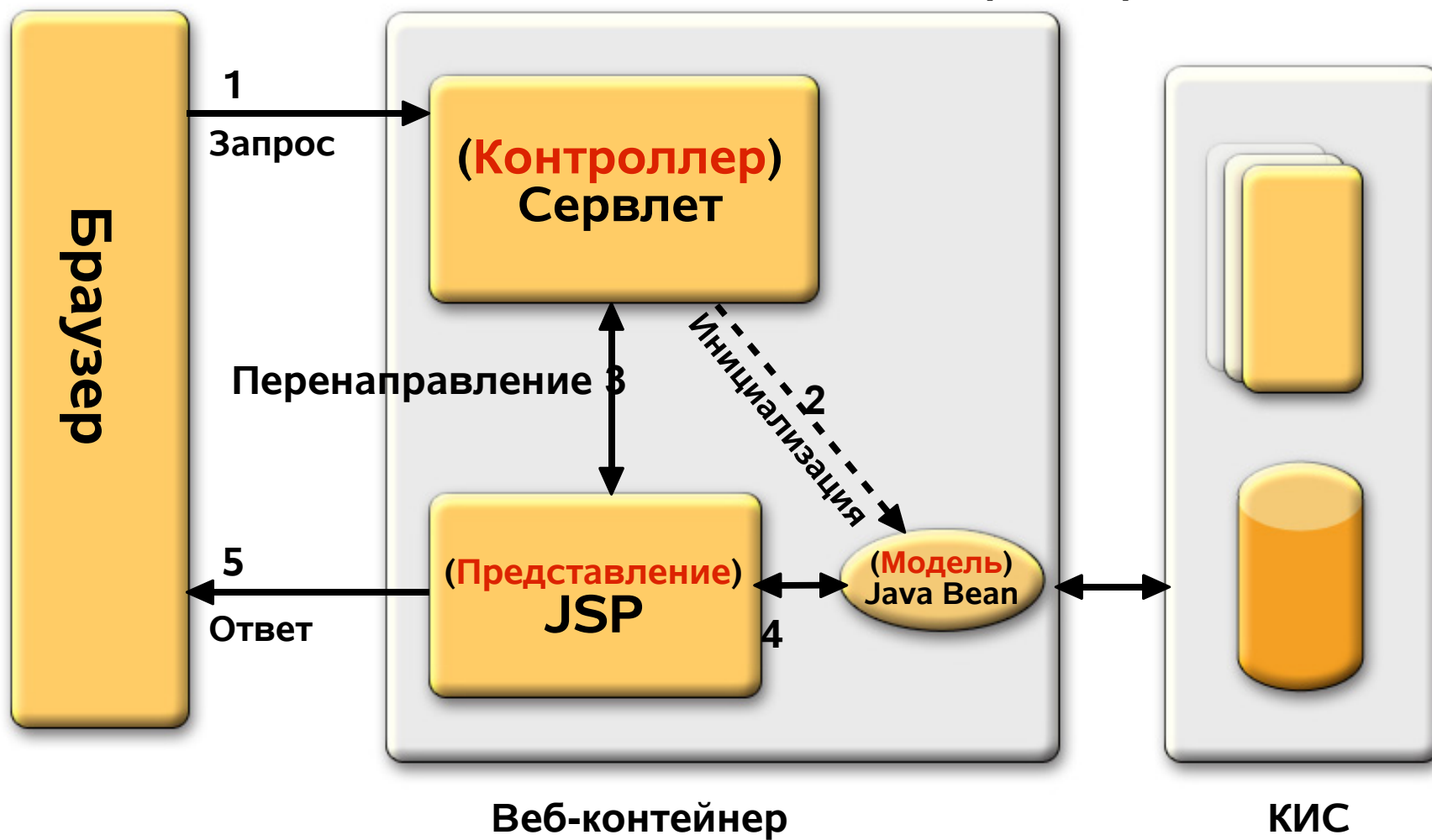


Архитектура Model 1 (на базе JSP-страниц)

- Состоит из множества взаимосвязанных JSP-страниц
 - > JSP-страницы выполняют все аспекты приложения - представление, контроллер и бизнес-логику
- Бизнес-логика и логика контроллера жестко запрограммированы **в JSP-страницах**
 - > В виде JavaBean-ов, скриптлетов, выражений
- Определение следующей страницы:
 - > Пользователь жмет на ссылку, ``
 - > Действие при отправке формы, `<FORM ACTION="search.jsp">`

Архитектура Model 2 (на базе сервлета)

Шаблон проектирования MVC

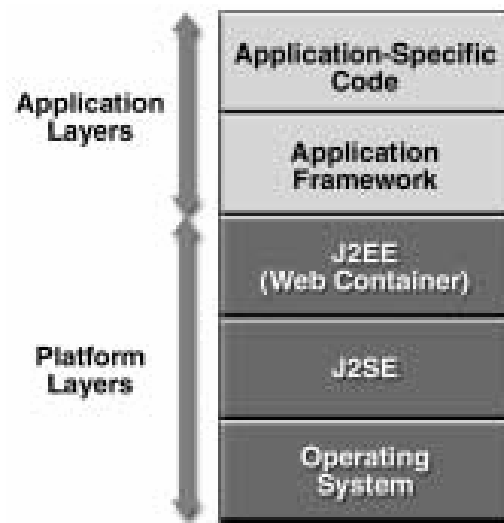


Архитектура Model 2 (на базе сервлета)

- JSP-страницы используются только для **представления**
 - > Управление и бизнес-логику выполняет сервлет (или несколько сервлетов)
- Сервлет является **единственной точкой входа**
 - > Обеспечивает общее управление безопасностью, журналированием, обработкой ошибок и т.п.
- Сервлет является **центральный контроллером**
 - > Преобразует входные данные в форму, используемую моделью
 - > Определяет бизнес-логику для обработки запроса
 - > Выполняет перенаправление на представление

Каркас веб-приложения (Web application framework)

- Основан на архитектуре Model 2
- Реализует набор базовых требований, общих для всех веб-приложений:
 - > определение операции по запросу
 - > вызов методов модели
 - > выбор отображаемого представления
 - > (сборка представления из шаблонов)
- Предоставляет классы и интерфейсы, которые разработчики могут использовать/расширять
- *Apache Struts, Tapestry, JavaServer Faces*



Достоинства каркаса веб-приложения

- Разделение представления и логики по различным компонентам
- Специализация труда разработчиков
- Обеспечение единой точки управления
- Упрощение тестирования и сопровождения модулей
- Поощрение разработки и использования стандартизованных компонентов
- Стабильность базовых классов и интерфейсов каркаса
- Поддержка сообществом, в том числе исправление ошибок, создание обучающих материалов, консультирование, разработка полезных дополнений