Стефан Хаас, Бинс Мэтью

Модель программирования ABAP RESTful

Разработка ABAP для SAP S/4HANA

ЧАСТЬ I SAP S/4HANA и SAP Fiori

# Глава 1 Архитектура SAP S/4HANA

*В этой главе рассматриваются технические строительные блоки SAP S/4HANA и описывается, как они взаимодействуют друг с другом. Вы узнаете об основных технических различиях между SAP S/4HANA и SAP Business Suite, а также изучить различия между SAP S/4HANA и SAP S/4HANA Cloud.*

Давайте начнем с общего обзора базовой архитектуры системы и ее технических строительных блоков. Мы более подробно рассмотрим в следующем разделе, где мы рассмотрим виртуальную модель данных SAP S/4HANA (VDM) и лежащую в ее основе технологию ABAP, основные службы данных (CDS). VDM также является основой транзакционных и аналитических приложений в SAP S/4HANA, и мы рассмотрим архитектуры этих типов приложений в следующих разделах. Кроме того, мы рассмотрим SAP S/4HANA и SAP S/4HANA Cloud и их различия. Мы завершим главу кратким введением в SAP Cloud Platform как основную платформу для разработки параллельных расширений для SAP S/4HANA.

## 1.1 Базовая архитектура

SAP S/4HANA — это программное обеспечение SAP для планирования ресурсов предприятия (ERP) последнего поколения, которое было впервые запущено в марте 2015 года. Как и предыдущее решение ERP, SAP ERP 6.0 Enhancement Pack (EHP) 8, который также служит основой для SAP S/ 4HANA предлагает интегрированный набор приложений, охватывающих ключевые бизнес-функции организации, включая финансовый учет (FI), контроллинг (CO) и учет основных средств (AA) в продажах и распределении (SD), управление материальными потоками (MM), управление продуктами Планирование (PP), управление качеством (QM), техническое обслуживание завода (PM) и человеческие ресурсы (HR).

Тем не менее, технологический сдвиг между классической SAP ERP и SAP S/4HANA огромен, поскольку SAP S/4HANA полностью построена на платформе SAP для вычислений в памяти, SAP HANA, и предлагает совершенно новый потребительский пользовательский интерфейс (UX) с SAP Fiori. В целом SAP пытается устранить избыточность и упростить систему, которая разрослась за последние десятилетия и во многих местах разошлась в чрезмерно сложные транзакции с использованием разных структур данных и разных архитектур. Во многих случаях сложные транзакции уже были заменены несколькими новыми ролевыми приложениями SAP Fiori, чтобы уменьшить сложность для конечных пользователей и улучшить UX в целом, что будет продолжаться и в будущем. Кроме того, эксклюзивное использование SAP HANA позволило некоторым модулям приложений упростить свои модели данных и сократить объем данных, поскольку SAP HANA значительно ускоряет создание отчетов и делает ранее необходимые сводные таблицы устаревшими. Однако во многих случаях эти упрощения основаны на функциональных стратегиях, уже существующих в предыдущих версиях ERP, что упрощает переход на SAP S/4HANA.

SAP периодически публикует список упрощений для SAP S/4HANA. В настоящее время SAP предлагает три сценария миграции на SAP S/4HANA: преобразование системы, при котором уже существующая система SAP ERP преобразуется в систему SAP S/4HANA; новая реализация, которая представляет собой подход «с нуля»; или преобразование ландшафта, при котором отдельные приложения перемещаются в установку SAP S/4HANA для консолидации существующего ландшафта.

### 1.1.1 Обзор

Хотя SAP S/4HANA является самостоятельным продуктом, одной из его четких целей является сохранение совместимости с предыдущими выпусками SAP ERP для упрощения миграции. Поэтому почти все его приложения, настройки и системные данные хранятся в тех же таблицах базы данных, что и в предыдущих версиях SAP ERP. С точки зрения архитектуры SAP S/4HANA по-прежнему основана на хорошо зарекомендовавшей себя трехуровневой архитектуре клиент-сервер, как показано на рис. 1.1, при этом SAP HANA является единственной системой управления реляционными базами данных (RDMS), лежащей в основе системы. Три основных уровня системы, снизу вверх, — это уровень базы данных, состоящий из SAP HANA, сервера приложений SAP NetWeaver для ABAP (SAP NetWeaver AS для ABAP) и внешнего интерфейса SAP Fiori (SAP Gateway). Фронтенд-сервер SAP Fiori получает входящие клиентские HTTP-запросы для приложений SAP Fiori, работающих на панели запуска SAP Fiori и в браузере, а также для бизнес-данных через протокол открытых данных (OData), который использует HTTP в качестве протокола передачи данных. Классический клиент SAP GUI был заменен браузером в качестве единого клиентского приложения и среды выполнения пользовательского интерфейса (UI) системы, по крайней мере, для пользователей SAP S/4HANA Cloud. С другой стороны, в SAP S/4HANA вы по-прежнему можете получить доступ к классическим бэкэнд-транзакциям, но тогда вы упустите новые возможности.

Пользовательский интерфейс SAP Fiori. SAP NetWeaver AS для ABAP содержит несколько компонентов для обработки запросов клиентских приложений в зависимости от типа приложения.

Корпоративный поиск (ESH) поддерживает функцию поиска на панели запуска SAP Fiori (также известную как поиск SAP Fiori) и обеспечивает прямой общесистемный поисковый доступ к приложениям и бизнес-данным, хранящимся в SAP HANA. Аналитический механизм, входящий в состав SAP Business Warehouse (SAP BW), оценивает и выполняет аналитические запросы во время выполнения. Для построения приложений SAP Fiori end-to-end (E2E) модель программирования ABAP RESTful является предпочтительной платформой, начиная с SAP S/4HANA версии 1909. Запросы от уровня шлюза SAP направляются через уровень описания адаптации службы (SADL), и в зависимости от того, доступен ли запрос только для чтения или требуется доступ для записи, запрос передается либо в среду выполнения запроса, либо в среду выполнения бизнес-объекта соответственно.

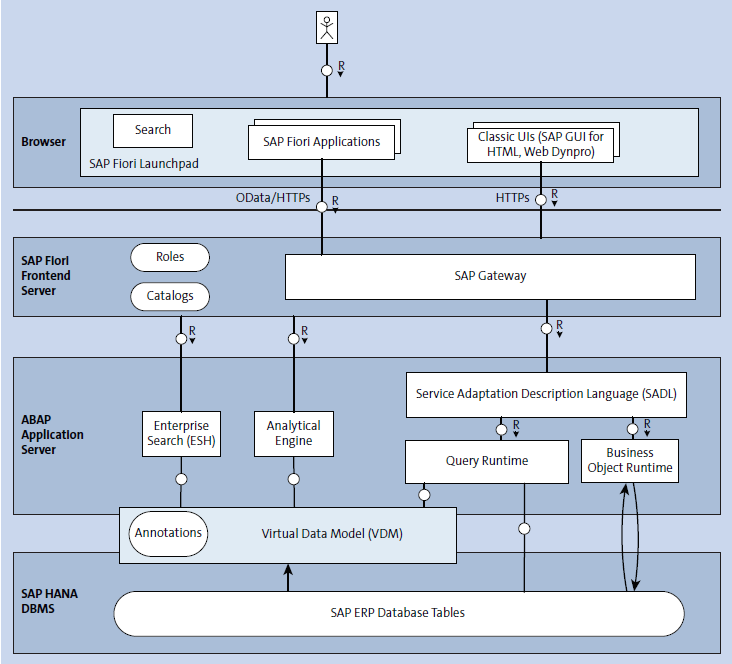


Рис. 1.1 Трехуровневая базовая архитектура SAP S/4HANA

### 1.1.2 Протокол открытых данных (OData)

С внедрением SAP Gateway и его канала OData SAP открыла ERP-систему для внешних систем, клиентов, устройств и машин. Все бизнес-объекты системы могут быть доступны по сети с помощью URI. Протокол OData предоставляет стандартизированный протокол для использования бизнес-сущностей и их взаимосвязей через парадигму RESTful, основанную на HTTP. Используя те же HTTP-команды GET, PUT, POST и DELETE, что и обычная служба RESTful, OData расширяет стандартную парадигму набором рекомендаций по созданию и использованию интерфейсов программирования приложений (API) RESTful. Например, OData добавляет стандартные параметры URL-адреса для ограничения запрошенного результирующего набора определенными свойствами ($select), определенным количеством сущностей ($top) или сущностями, удовлетворяющими определенному условию ($filter).

Можно сказать, что OData расширяет парадигму RESTful в сторону потребления данных, подобного SQL, через HTTP, что, в свою очередь, упрощает прямую отправку запросов в базу данных SAP HANA.

*Примечание*

*Чтобы узнать больше об Open Data Protocol (OData), посетите официальный веб-сайт https://www.odata.org/.*

### 1.1.3 SAP HANA

Естественно, одной из наиболее важных целей архитектуры SAP S/4HANA является полное использование возможностей SAP HANA в оперативной памяти. Чтобы способствовать этому использованию, SAP представила уровень VDM поверх существующих таблиц бизнес-данных, который позволяет семантически консолидировать все более и более фрагментированную модель данных с годами без потери совместимости с предыдущими версиями ERP. Основные службы данных ABAP (CDS) — это основная технология, позволяющая использовать VDM; CDS полностью интегрирован в стек ABAP, но выполняется в SAP HANA. Используя язык определения данных CDS (DDL), вы можете определять многоуровневые и семантически насыщенные представления данных, выбирая данные из старых таблиц ERP.

Технически, представления CDS будут генерировать представления базы данных SAP HANA при активации, отправляя вычисления и состав семантических бизнес-сущностей и предоставляя базу данных. Размещение согласованного семантически богатого виртуального уровня представления поверх старых таблиц базы данных абстрагирует сложные табличные структуры и отношения, которые развивались с течением времени, и обеспечивает согласованное и ориентированное на бизнес представление данных. Ориентированное на бизнес предоставление данных с поддержкой SAP HANA значительно повышает производительность приложений, поскольку расчет и предоставление бизнес-данных в лучшем случае полностью переносится в базу данных SAP HANA. Эта парадигма называется парадигмой преобразования кода в данные, в отличие от классической парадигмы преобразования данных в код, где большие наборы данных передаются на сервер приложений, а затем обрабатываются в ABAP для предоставления действительно необходимых бизнес-данных. Последняя парадигма, однако, никогда не достигнет полной производительности, которой может достичь SAP HANA.

Как технология SAP Basis, VDM и CDS являются основой всех типов приложений SAP S/4HANA, включая транзакционные приложения, аналитические приложения, внешние интерфейсы и корпоративный поиск (ESH). VDM — это прежде всего повторно используемая и стабильная база данных, используемая для внутренней разработки в SAP, но партнеры и клиенты также могут использовать и разрабатывать поверх явно выпущенных представлений. VDM увеличивает скорость и эффективность разработки, обеспечивая стандартизированное и простое использование бизнес-объектов.

*Примечание*

*Обзор представлений CDS, явно выпущенных SAP, см. в приложении View Browser, содержащемся в роли SAP\_BR\_ANALYTICS\_SPECIALIST.*

### 1.1.4 Сервер приложений SAP NetWeaver для ABAP

Хотя основное внимание уделяется эксплуатации SAP HANA и наблюдается тенденция к двухуровневой архитектуре, средний уровень архитектуры SAP S/4HANA по-прежнему состоит из сервера приложений SAP NetWeaver для ABAP (SAP NetWeaver AS для ABAP), как и в предыдущих выпусках SAP ERP. SAP NetWeaver AS для ABAP содержит несколько механизмов для обработки транзакционных и аналитических данных, например, Analytical Engine, который также является частью SAP Business Warehouse (SAP BW), Business Object Processing Framework (BOPF), среды выполнения бизнес-объектов и запросов, модели программирования ABAP RESTful и языка определения адаптации служб (SADL). В прошлом реализация бизнес-логики в SAP ERP была разнородной и фрагментарной, с использованием множества различных сред ABAP или вообще без них, что приводило к нестандартным приложениям, которые было сложно поддерживать и требовалось много знаний в предметной области. При первоначальном запуске SAP S/4HANA не было стандартизированной согласованной модели программирования для приложений SAP Fiori, а был только набор передовых методов, например, о том, как разработать службу OData на основе ABAP для канала OData (ODC) SAP-шлюз.

Чтобы стандартизировать реализацию бизнес-логики в SAP S/4HANA, SAP запустила модель программирования ABAP для SAP Fiori с SAP S/4HANA 1610 FPS01. С тех пор SAP рекомендует использовать эту модель программирования для E2E-разработки приложений SAP Fiori. В модели программирования базовая структура узла транзакционной платформы обработки бизнес-объектов (BOPF) получена из аннотированных сущностей CDS, а бизнес-логика реализована внутри BOPF в ABAP. Инфраструктура BOPF изначально была разработана для SAP Business By Design, а затем была повторно использована и интегрирована в модель программирования SAP Fiori для обработки частей бизнес-логики приложений на основе ABAP. Представления CDS и бизнес-логика BOPF могут быть представлены на основе модели либо через проект построителя сервисов SAP Gateway (транзакция SEGW), просто сославшись на объекты CDS, которые должны быть представлены (сценарий эталонного источника данных), либо через аннотацию в корневом представлении модели данных CDS (*@OData.publish: true*).

Однако модель программирования ABAP для SAP Fiori по-прежнему представляет собой разнородный набор новых и существующих технологий и инструментов. Для определенных действий требуется даже доступ к SAP GUI, например, для создания службы OData с помощью транзакции SEGW или для активации службы OData с помощью транзакции /IWFND/MAINT\_SERVICE. Чтобы преодолеть эти недостатки и предоставить готовую к облаку модель программирования, в августе 2018 года SAP выпустила модель программирования ABAP RESTful для среды ABAP (платформа ABAP 1808). С тех пор среда ABAP стала частью SAP Cloud Platform как предложение «Платформа как услуга» (PaaS), которое предоставляет опыт разработки E2E для оптимизированных для SAP HANA сервисов OData, таких как приложения SAP Fiori. Начиная с SAP S/4HANA 1909, модель программирования ABAP RESTful также доступна в локальных установках SAP S/4HANA и заменяет модель программирования ABAP для SAP Fiori. Как и в предыдущей модели программирования ABAP для SAP Fiori, семантически богатая модель данных приложений формируется представлениями CDS. Для обработки транзакций бизнес-объекты получаются из сущностей CDS; какие операции поддерживает бизнес-объект (например, операции создания, чтения, обновления и удаления (CRUD) и настраиваемые действия), определяется с помощью, так называемого определения поведения. Определение поведения — это новый объект репозитория ABAP, определенный с использованием языка определения поведения (BDL). Затем модель данных и ее поведение предоставляются как бизнес-служба, которая по сути является службой RESTful OData. Бизнес-служба состоит из определения службы и привязки службы. Определение службы — это объект репозитория ABAP, который определяет, какие части модели данных должны быть представлены в виде бизнес-службы. Привязка службы — это объект репозитория ABAP, используемый для привязки определения службы к определенному протоколу связи клиент-сервер, например OData версии 2. В отличие от модели программирования ABAP для SAP Fiori, ***новая модель программирования ABAP RESTful была напрямую интегрирована в язык ABAP*** с новыми переносимыми ***объектами разработки: определение поведения, определение службы и привязка службы***. Эта интеграция обеспечивает согласованный опыт разработки E2E на основе Eclipse и позволяет использовать модель программирования в облаке без каких-либо дополнительных необходимых платформ и инструментов, а только с ядром языка ABAP. На рис. 1.2 показана эволюция разработки приложений на платформе ABAP в целом и, в частности, на SAP S/4HANA.

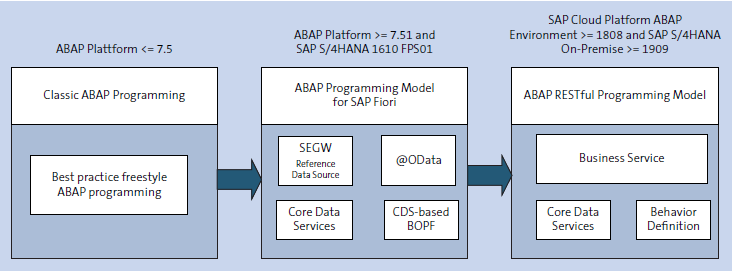


Рис. 1.2 Эволюция модели программирования ABAP в SAP S/4HANA

### 1.1.5 Интерфейсный сервер SAP Fiori на базе SAP NetWeaver Gateway

Фронтенд-сервер SAP Fiori основан на компоненте SAP Gateway SAP NetWeaver и управляет связью без сохранения состояния между браузером пользователя в качестве клиентского приложения системы и бэкэндом. На сервере размещаются приложения SAP Fiori, которые загружаются для обработки на стороне клиента в браузере пользователя, а также службы или API-интерфейсы OData, используемые для получения или изменения бизнес-данных, отображаемых в приложениях. В собственных приложениях SAP Fiori существует четкое разделение между пользовательским интерфейсом и данными, что способствует повторному использованию серверных компонентов, будь то бизнес-логика в представлениях ABAP или CDS. Компонент SAP Gateway реализует протокол OData через свой канал OData и заботится обо всех задачах, связанных с протоколом, например, об анализе и проверке запросов, отображении формата ответа в формате JSON или XML, выполнении входящих и исходящих преобразований и многом другом. Интерфейсный и внутренний сервер взаимодействуют друг с другом через соединение доверенного удаленного вызова функций (RFC).

SAP Gateway поддерживает развертывание концентратора, а также встроенное развертывание. В сценарии узлового развертывания, который раньше был рекомендуемой настройкой для SAP S/4HANA до SAP S/4HANA 1809, компоненты SAP Gateway развертываются в автономной внешней системе и не связаны с внутренней системой SAP S/4HANA. Эта настройка, среди прочего, повышает безопасность, поскольку внешняя поверхность атаки не включает серверную систему. С другой стороны, в сценарии встроенного развертывания, который является рекомендуемым вариантом развертывания, начиная с SAP S/4HANA 1809, и шлюз SAP, и серверные компоненты находятся в одной системе, что снижает общую стоимость владения (TCO). Однако этот сценарий увеличивает внешнюю поверхность атаки, а циклы обновления для внешнего и внутреннего серверов связаны. Однако интерфейс SAP Fiori зависит от серверной части SAP S/4HANA и наоборот (например, для SAP Fiori для SAP S/4HANA 1809 требуется серверная часть с SAP S/4HANA 1809). Поэтому теперь мы рекомендуем развернуть интерфейс SAP Fiori и серверную часть SAP S/4HANA в одной системе, чтобы обеспечить их синхронизацию.

### 1.1.6 SAPUI5

Уровень пользовательского интерфейса (UI) состоит из собственных приложений SAP Fiori, разработанных в SAPUI5, который представляет собой среду JavaScript, позволяющую разрабатывать адаптивные приложения HTML5 в соответствии с языком дизайна SAP Fiori. Приложения, разработанные с помощью SAPUI5, запускаются в браузере и, по своей природе отзывчивые, меняют свой макет в зависимости от размера экрана или разрешения клиентского устройства, на котором работает браузер. В дополнение к собственным приложениям SAP Fiori в SAP S/4HANA также поддерживаются более старые технологии пользовательского интерфейса SAP, поддерживающие HTML-подобный Web Dynpro или SAP GUI для HTML, и они получат те же темы, что и собственные приложения SAP Fiori при запуске на панели запуска SAP Fiori. Для более старых технологий пользовательского интерфейса пользовательский интерфейс не отображается на стороне клиента, и не происходит разделения пользовательского интерфейса и API-интерфейсов бизнес-данных. Вместо этого пользовательский интерфейс отображается на стороне сервера, и, поскольку эти приложения были разработаны исключительно для настольных компьютеров, классические пользовательские интерфейсы не такие отзывчивые.

### 1.1.7 Панель запуска SAP Fiori

Конечные пользователи могут получить доступ к SAP S/4HANA через браузеры, а единственной точкой входа в систему является панель запуска SAP Fiori. На домашней странице панели запуска SAP Fiori отображается сетка плиток, которые обеспечивают доступ к приложениям, назначенным текущему пользователю. Приложения SAP Fiori организованы в каталоги, которые создаются в конструкторе панели запуска SAP Fiori и позже назначаются ролям транзакции ***PFCG***. Затем эти роли, в свою очередь, назначаются бизнес-пользователям и включают в себя все приложения, необходимые пользователю для выполнения его повседневных бизнес-задач. Панель запуска SAP Fiori также предоставляет встроенную функцию поиска в области заголовка на основе ESH, также известную как поиск SAP Fiori, с помощью которой пользователь может выполнять поиск по всем существующим и поддерживающим ESH бизнес-объектам в системе.

## 1.2 Серверная часть

В следующих подразделах мы подробно рассмотрим серверные компоненты и технологии SAP S/4HANA. Прежде всего, мы рассмотрим VDM SAP S/4HANA, который поддерживается технологией CDS и поддерживает парадигму преобразования кода в данные и передачу кода в базу данных SAP HANA. ***VDM*** является основой для всех типов приложений в SAP S/4HANA и дополнена ***BOPF*** (в версиях < чем SAP S/4HANA 1909) или ***определениями поведения*** (начиная с версии SAP S/4HANA 1909) для транзакционных приложений и ***аналитическим механизмом*** для аналитических приложений. В следующих подразделах мы дадим общее представление о различных типах приложений SAP S/4HANA, а также о том, какие технологии и компоненты архитектуры используются при разработке различных приложений для SAP S/4HANA.

### 1.2.1 Виртуальная модель данных (VDM)

Помимо классических таблиц SAP ERP, VDM является основным архитектурным изменением SAP S/4HANA по сравнению с предыдущими выпусками SAP ERP, как показано на рис. 1.3.

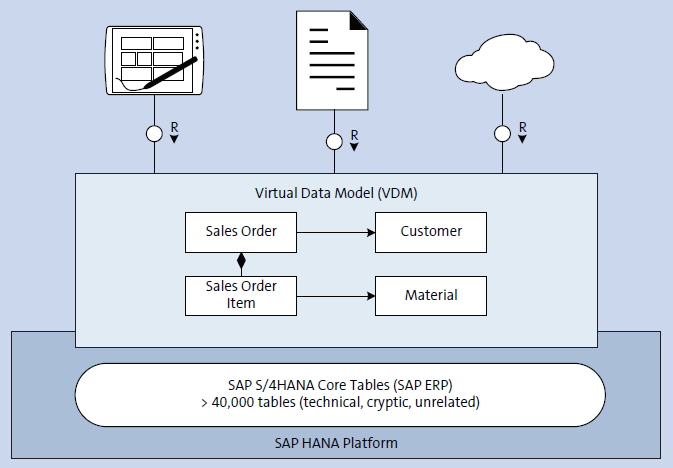


Рисунок 1.3 VDM поверх старых основных таблиц SAP ERP как краеугольный камень всех типов приложений SAP S/4HANA

VDM абстрагирует сложные и гетерогенно разросшиеся структуры таблиц, помещая упрощенный и ориентированный на бизнес уровень представлений CDS поверх старых таблиц базы данных. VDM обеспечивает понятный, семантически богатый и упрощенный доступ к бизнес-данным в форме используемых бизнес-сущностей, при этом потребителям не нужно разбираться в сложных структурах таблиц и взаимосвязях внешних ключей, которые накапливались десятилетиями в таблицах базы данных.

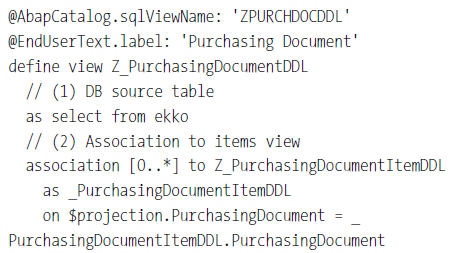
#### Базовые службы данных ABAP CDS

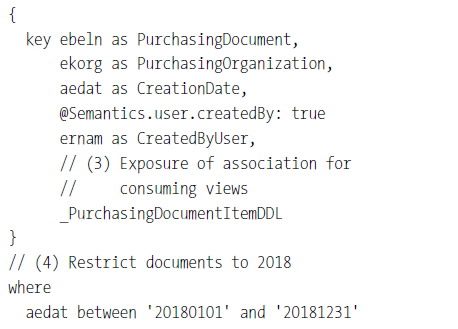
Технологическим ядром VDM является технология CDS, состоящая из DDL, языка запросов и языка управления данными (DCL). Артефакты CDS — это новые объекты разработки в стеке ABAP, которые используют стандартную систему управления жизненным циклом транспорта ABAP, хотя эти объекты разработки выполняются в SAP HANA.

Язык определения данных (DDL) можно рассматривать как усовершенствование SQL, и он позволяет определять семантически насыщенные представления базы данных, выбранные либо из таблиц базы данных, либо из других представлений CDS. CDS DDL доступен начиная с ABAP 7.4 с пакетом обновления (SP) 2. Отношения между представлениями выражаются с помощью концепции, называемой ассоциациями. DDL также может содержать аннотации для добавления метаданных домена.

В листинге 1.1 показан упрощенный пример CDS DDL. Представление выбирает данные из таблицы EKKO, которая представляет собой классическую таблицу SAP ERP, содержащую документы закупок и принадлежащую прикладному компоненту Управление материальными потоками и закупками (MM-PUR). Как видите, синтаксис напоминает SQL. Код включает часть прогноза, в которую мы включили несколько полей базовой таблицы, и часть выбора (условие WHERE), где мы ограничили результаты документами, созданными в 2018 году. В части прогноза вы можете скрыть зашифровывать имена полей таблицы технической базы данных и предоставлять семантически значимые имена с использованием псевдонимов, что также делается в SAP S/4HANA VDM; например, в нашем примере псевдоним PurchasingDocument был определен для поля базы данных ebeln.

Ассоциации используются для выражения отношений между различными представлениями CDS и напоминают объединения, но их легче читать, поскольку они не программируют соединения напрямую. При выборе данных из зависимых представлений с использованием выражений пути этих соединений будут материализованы; в противном случае ассоциации можно рассматривать как метаданные, и между связанными представлениями не выполняются соединения. В последней строке проекции мы распространяем ассоциацию от документов покупки к элементам документов покупки другим представлениям потенциального потребления (\_PurchasingDocumentItemDDL). Затем потребляющее представление могло бы собирать данные из документов, а также элементов и формировать новый бизнес-объект; например, выражение \_PurchasingDocumentItemDDL. PurchasPurchasing-DocumentItem выберет поле элемента документа покупки представления Z\_Purchasing-DocumentItemDDL через представление Z\_PurchasingDocumentDDL, что приведет к преобразованию ассоциации в фактическое объединение документов и товаров покупки.





Листинг 1.1. Простой пример DDL-файла в верхней части таблицы закупочных документов EKKO

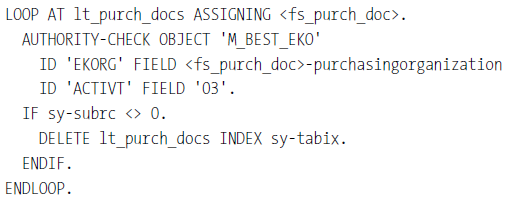
Представления CDS можно использовать через Open SQL или другие представления CDS; поэтому Open SQL или CDS сами формируют языки запросов CDS, как показано в листинге 1.2.



Листинг 1.2. Открытие SQL, выбранного в ABAP, для выбора данных из представления CDS документа о покупке

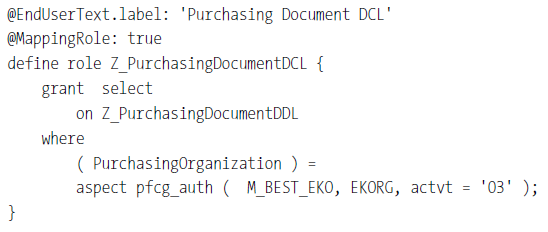
Доступ к данным через представления CDS можно ограничить с помощью языка управления данными (DCL) CDS. В файле DCL авторизации объявляются через стандартные объекты авторизации ABAP. Во время выполнения объекты авторизации пользователя (транзакции PFCG) и их значения будут собираться и передаваться в базу данных в качестве условий фильтрации. Эта чрезвычайно важная функция продвигает парадигму Code-to-Data. Раньше для проверки прав доступа к отдельным записям базы данных или уровням экземпляров данные приходилось передавать на сервер приложений. Затем отдельные записи проверялись на основе значений, найденных в назначенных пользователю объектах авторизации.

В листинге 1.3 показано, как этот процесс авторизации мог работать в прошлом. Данные выбираются из базы данных с помощью Open SQL и сохраняются во внутренней таблице lt\_purch\_docs. В цикле проверяется каждая запись, чтобы определить, имеет ли пользователь право видеть запись; в противном случае запись удаляется из таблицы перед передачей в пользовательский интерфейс для отображения. Однако в результате мы теряем преимущество в производительности Data-to-Code, полученное с CDS DDL, потому что только часть выбора данных передается в SAP HANA, тогда как необходимая часть проверки авторизации по-прежнему выполняется в ABAP и требует передачи потенциально больших наборов данных из базы данных на сервер приложений по сети.



Листинг 1.3. Классические проверки AUTHORITY-CHECK в ABAP после чтения данных из базы данных с использованием Open SQL и сохранения во внутренней таблице (lt\_purch\_docs)

К счастью, начиная с ABAP 7.40 SP 10, файлы CDS DCL можно использовать для отправки этих проверок авторизации на уровне экземпляра в базу данных. В листинге 1.4 показан простой файл DCL для представления Z\_PurchasingDocumentDDL. Как и в предыдущем фрагменте кода ABAP, этот код использует стандартный объект авторизации ABAP M\_BEST\_EKO для ограничения доступа к закупочным документам закупочной организацией, как показано на рис. 1.4. Закупочные организации, для которых пользователю разрешено просматривать закупочные документы, являются производными от ролей транзакции PFCG. Пользователю назначаются роли в PFCG, которые, в свою очередь, содержат объекты авторизации, имеющие определенные значения полей авторизации, специально необходимые для выполнения конкретной бизнес-задачи пользователя в соответствии с принципом наименьших привилегий. В следующем DCL мы позаботимся о том, чтобы пользователи могли видеть только записи документов закупок с PurchasingOrganization, которые также содержатся в поле EKORG экземпляра объекта авторизации M\_BEST\_EKO, назначенного пользователю через роль в PFCG, например закупочные организации 0001 и 0002. Кроме того, мы можем проверить право на отображение, используя поле acvt.



Листинг 1.4. Простой файл DCL с использованием стандартного объекта авторизации ABAP M\_BEST\_EKO для фильтрации закупочных документов по закупочной организации

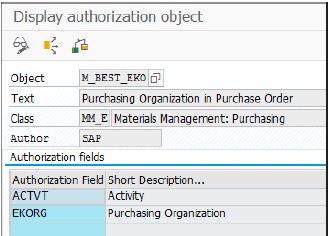


Рисунок 1.4 Объект авторизации для закупочных организаций в заказах на поставку

Во время выполнения исполняемый оператор SQL SELECT будет выглядеть следующим образом:



Обратите внимание, как проверки авторизации DCL прозрачно передаются в базу данных с помощью предложения SQL WHERE.

#### Архитектура виртуальной модели данных (VDM)

Чтобы стимулировать повторное использование, сохраняя при этом оптимально адаптированные представления CDS для различных сценариев использования и приложений, SAP организовала стандартные представления SAP S/4HANA CDS в многоуровневой архитектуре. Основу архитектуры VDM составляют старые базовые таблицы SAP ERP для различных компонентов приложений, от FI до Logistics и HR. На рис. 1.5 показаны различные уровни архитектуры VDM. Снизу вверх архитектура состоит из слоя интерфейса, образованного ***базовыми (basic)*** и ***составными (composite)*** представлениями, и слоя представлений ***потребления (consumption)***, который находится поверх слоя представления ***интерфейса (external API)***.

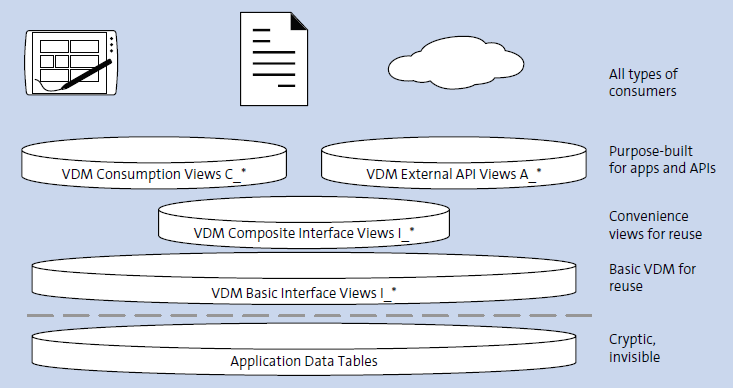


Рисунок 1.5 Различные уровни архитектуры VDM

Самый нижний уровень представлений CDS поверх таблиц базы данных называется ***уровнем представлений интерфейса***, который служит повторно используемым базовым уровнем VDM, не зависящим от приложений. Уровень представления интерфейса можно дополнительно разделить на ***базовое*** и ***составное*** представления. По соглашению имена представлений CDS уровня представлений интерфейса, как базовых, так и составных, начинаются с ***I\_***, например, I\_Material. Различие между базовыми и составными представлениями осуществляется с помощью аннотации заголовка:

***@VDM.viewType: #BASIC/#COMPOSITE***.

***Базовый*** уровень представления представлений CDS является базовым уровнем самого VDM. Предполагается, что все представления CDS верхнего уровня VDM, как составные, так и потребления, предназначены для создания бизнес-сущностей, специфичных для приложения, и доступа к данным с использованием этих базовых представлений CDS. Однако прямой доступ к таблице базы данных представлений CDS верхнего уровня является плохой практикой и запрещен. Многие другие правила применяются к разработке основных представлений CDS, но основные аспекты и характеристики, которые вы должны отметить, следующие:

- Базовые представления обычно выбирают данные из одной таблицы базы данных, чтобы обеспечить повторное использование с высокой степенью детализации.

- Если обязательные атрибуты объекта распределены по нескольким таблицам, базовое представление может уже объединить их вместе.

- Базовые представления могут иметь ассоциации только с другими базовыми представлениями.

- Базовые представления не должны содержать никаких сложных вычислений, агрегаций, параметров или условий WHERE — только проекция полей базовой таблицы базы данных.

- Базовые представления предоставляют семантические имена полей, чтобы скрыть технические сокращенные имена полей таблицы базы данных.

- Базовые взгляды не зависят от их потенциальных потребителей.

- Базовые представления не содержат избыточности, что означает, что часть данных может быть выбрана только с использованием одного конкретного основного представления (например, существует ровно одно представление для объекта рабочего центра или центра затрат).

Благодаря своей простоте базовые представления CDS являются оптимальными кандидатами для многократного повторного использования и составления новых, более специфичных для приложения представлений и сущностей. Представления CDS уже предоставляют основные бизнес-объекты без избыточности, несущие семантические имена и семантически помеченные свойства, но они по-прежнему обычно основаны на взаимно-однозначных отношениях между базовым представлением и таблицей SAP ERP.

Для создания новых повторно используемых бизнес-объектов или объединения всех бизнес-данных, необходимых для определенного существующего бизнес-объекта, в одном представлении предоставляется слой ***составного*** представления, который находится поверх уровня базового представления. Чем дальше мы продвигаемся вверх по различным уровням, тем более специфичными для приложения и бизнес-ориентированными становятся представления. Основные характеристики составных видов следующие:

- Составные представления основаны на одном или нескольких базовых представлениях CDS и могут также выбирать данные из других составных представлений.

- Составные представления могут иметь ассоциации с другими составными и базовыми представлениями.

- Составные представления могут содержать вычисления, агрегации, параметры или условия WHERE.

- По своей природе составные представления создают избыточность данных; например, нам может понадобиться многократно используемое представление для аналитического варианта использования и для транзакционного приложения, которое может возвращать одни и те же данные.

Как и базовые представления, составные представления предназначены для повторного использования. Однако, в отличие от базовых представлений, составные представления могут уже реализовывать сценарии для конкретных типов приложений, такие как определенные вычисления, агрегации, аннотации или сценарии объединения, чтобы предоставлять бизнес-сущности и данные для представлений потребления для конкретных приложений, расположенных поверх них.

Самый верхний уровень архитектуры VDM, уровень представления ***потребления***, предназначен для конкретных приложений. В отличие от слоев представления интерфейса, представления потребления не предназначены для повторного использования; вместо этого они обращаются к конкретным требованиям приложения. По соглашению их имена начинаются с ***C\_***, например, C\_MaterialDailyDemand. Представления уровня потребления дополнительно снабжены аннотацией заголовка ***@VDM.viewType: #CONSUMPTION***.

***Внешние представления API***, поскольку они специфичны для приложения и не могут использоваться повторно, также назначаются слою представления потребления и являются основой для опубликованных API-интерфейсов OData, например, через SAP API Business Hub. Их имена начинаются с ***A\_***, например, A\_ProductionOrder.

Основные характеристики представлений слоя ***потребления*** следующие:

- Представления уровня потребления учитывают требования конкретного приложения.

- Представления потребления не предназначены для повторного использования, поскольку они разрабатываются для определенной цели, например, для определенного приложения SAP Fiori или аналитического запроса.

- Представления потребления должны получать доступ к данным, используя представления уровня интерфейса.

- Представления потребления могут иметь ассоциации с представлениями интерфейса, а также с другими представлениями потребления.

Представления потребления могут содержать специфичные для приложения сложные вычисления, агрегации, условия соединения или аннотации пользовательского интерфейса, что делает невозможным их повторное использование вне границ приложения.

На рис. 1.6 показан полный пример E2E VDM приложения на основе объекта заказа на продажу, который является частью компонента приложения SD.

Самый нижний уровень содержит различные таблицы SAP ERP, из которых данные объединяются с использованием нескольких уровней представления CDS. Слой базового представления (I\_SalesOrder, I\_SalesOrderItem, I\_Material, I\_MaterialText) напрямую выполняет проекции на таблицы базы данных, тогда как слой составного представления обеспечивает многоразовые представления для конкретных приложений: транзакционных (I\_SalesOrderTP) или аналитических приложений (I\_SalesOrderItemCube). Наконец, уровень представления потребления состоит из представлений для конкретных приложений: транзакционных (C\_SalesOrderTP), аналитических (C\_SalesOrderItemQuery) и внешних приложений API (A\_SalesOrderItem).

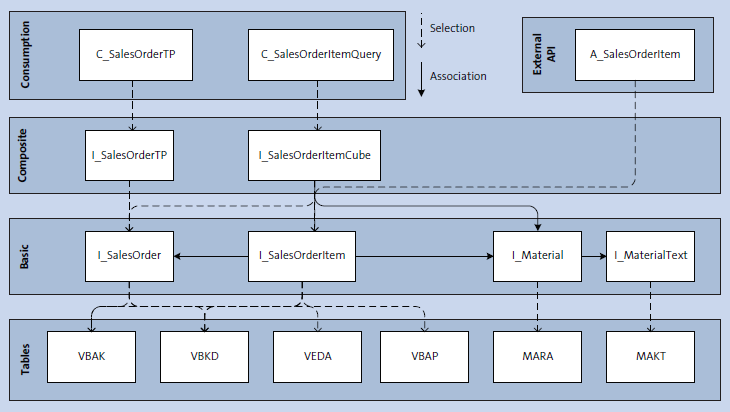


Рис. 1.6 Пример архитектуры VDM для бизнес-объекта заказа на продажу

### 1.2.2 Модель программирования ABAP для SAP Fiori

Как упоминалось в разделе 1.1.4, модель программирования ABAP для SAP Fiori является рекомендуемой моделью программирования для транзакционных приложений вплоть до SAP S/4HANA 1809 (первоначальный выпуск в сентябре 2018 г.). Таким образом, в центре внимания этой книги — новая модель программирования ABAP RESTful, доступная начиная с SAP S/4HANA 1909 (первоначальный выпуск в сентябре 2019 г.). Тем не менее, в этом разделе мы дадим краткое введение в модель программирования ABAP для SAP Fiori, так как вы наверняка столкнетесь со многими приложениями в вашей системе SAP S/4HANA, использующими эту модель программирования.

*Примечание*

*Чтобы узнать больше о модели программирования ABAP для SAP Fiori, ознакомьтесь с другой нашей книгой SAP PRESS: Модель программирования ABAP для SAP Fiori: разработка ABAP для SAP S/4HANA по адресу* [*http://www.sap-press.com/4766*](http://www.sap-press.com/4766).

Неотъемлемой частью модели программирования ABAP для SAP Fiori является BOPF, структура узлов которого основана на аннотированных ассоциациях представлений CDS. Обратите внимание, что хотя BOPF добавляет только бизнес-логику приложения, базовая модель данных всегда получается из VDM с использованием конкретных аннотаций иерархии для моделирования структуры узла BOPF на уровне CDS. Создание соответствующей структуры узла BOPF из бизнес-сущностей CDS поддерживается, начиная с версии SAP S/4HANA 1610 SPS 03.

Общая идея модели программирования ABAP для SAP Fiori в SAP S/4HANA, как показано на рис. 1.7, заключается в предоставлении прямого высокопроизводительного доступа к представлениям CDS для сценариев только для чтения с использованием инфраструктуры SADL, которая прозрачно передает запросы OData через SAP Gateway вниз к уровню VDM, используя свою среду выполнения запросов. Доступ к модификации предоставляется через среду выполнения BOPF для вызова бизнес-логики. Если изменение доступа вообще не вызывает никакой бизнес-логики, дальнейшая оптимизация будет заключаться в прямом доступе к таблицам базы данных, что предотвращает накопление локальных буферов и кэшей.

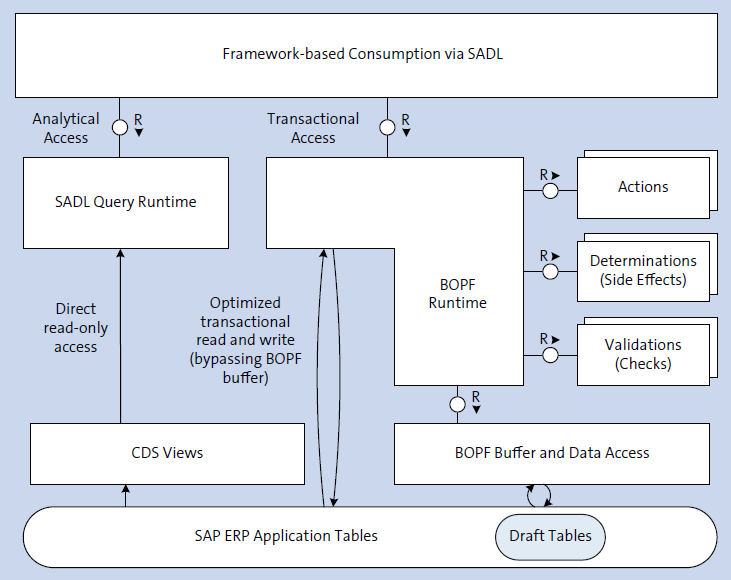


Рис. 1.7 Архитектура модели программирования ABAP для SAP Fiori

Пользовательский интерфейс SAP Fiori требует разделения интерфейса и сервера, а также реализации сервера, оптимизированной для обработки запросов без сохранения состояния. До сих пор BOPF был нацелен на бизнес-приложения с отслеживанием состояния, но было приложено много усилий для оптимизации BOPF для обработки данных без сохранения состояния. Классические приложения SAP Dynpro или Web Dynpro полагаются на сеанс ABAP вместе с буферами приложений, которые обслуживают клиентские запросы до тех пор, пока пользователь не завершит свою работу и не сохранит все изменения данных. Приложения SAP Fiori, с другой стороны, извлекают данные с помощью служб RESTful OData, которые не имеют состояния, как HTTP, который используется в качестве протокола передачи данных. Отсутствие состояния также является основным свойством парадигмы RESTful и имеет некоторые преимущества по сравнению с приложениями с состоянием:

- Отсутствие состояния обеспечивает горизонтальное масштабирование API, поскольку любой запрос может быть обработан любым сервером без каких-либо зависимостей, связанных с сеансом.

- API без сохранения состояния менее сложны, так как вся информация, необходимая для обслуживания запроса, содержится в текущем запросе, поэтому логика синхронизации состояния на стороне сервера не требуется.

- Надежность также повышена, потому что в случае сбоя одного сервера другой сервер может немедленно взять на себя управление, поскольку для обслуживания запросов не требуется ранее сохраненное состояние.

- Для повышения эффективности сети ответы на запросы могут кэшироваться на стороне клиента.

Вы также можете реализовать сценарий приложения с отслеживанием состояния, используя связь без сохранения состояния и BOPF, но вместо хранения данных в контексте сеанса ABAP в памяти данные должны храниться в базе данных с использованием черновых таблиц, которые являются теневыми таблицами исходных таблиц базы данных. Пользователи могут работать с предварительными данными в течение нескольких шагов взаимодействия и циклов, а после завершения редактирования объекта черновые данные могут быть активированы и записаны в фактические таблицы базы данных.

Когда пользователь изменяет бизнес-данные и переключается между полями ввода на экране, изменения немедленно отправляются в серверную часть через запрос OData, который обновляет черновые таблицы вместо исходных таблиц базы данных. Каждое черновое обновление, отправляемое серверной части, называется двусторонним.

Эта информация чернового объекта, которая присутствует для определенного бизнес-объекта, также может использоваться для блокировки объекта для других пользователей (монопольная блокировка). В прошлом блокировки были привязаны к сеансу ABAP; теперь вы можете использовать надежные блокировки, чтобы включить блокировку объекта без сохранения состояния за пределами нескольких сеансов и запросов ABAP.

#### Реализация бизнес-логики

Реализация бизнес-логики в BOPF осуществляется в виде действий, определений или проверок:

- Действия (Actions)

Действия реализуют операции, например, изменение статуса счета с «Ожидается» на «Оплачен».

- Валидации (Validations)

Валидации не изменяют никаких данных, но проверяют, согласуются ли данные, например, определены ли цена и количество для счета-фактуры.

- Определения (Determinations)

Определения имеют побочные эффекты и инициируются действиями, например, если количество товаров в счете-фактуре изменяется, общая цена также должна быть обновлена.

#### Сценарии транзакционных приложений

В целом модель программирования ABAP для SAP Fiori, доступная начиная с SAP S/4HANA 1610 SPS 03, поддерживает четыре различных сценария транзакционных приложений на основе BOPF и CDS:

- Новое приложение с возможностью черновика

- Новое приложение без черновика

- Проект включения существующего приложения

- Приложение только для чтения с быстрыми действиями (BOPF используется только для действий, но не для операций CRUD)

### 1.2.3 Модель программирования ABAP RESTful

Как упоминалось ранее, основное внимание в этой книге уделяется новой модели программирования ABAP RESTful, которая заменяет ранее описанную модель программирования ABAP для SAP Fiori с помощью SAP S/4HANA 1909. Кроме того, SAP пообещал поддержку перехода на модель программирования ABAP RESTful для клиентов, которые уже вложили средства в модель программирования ABAP для SAP Fiori. По возможности, с этого момента вы должны использовать модель программирования ABAP RESTful для разработки новых приложений, поскольку планируется, что модель программирования ABAP RESTful станет долгосрочным решением для разработки сервисов RESTful OData на основе ABAP как в облаке, так и в локальной среде установки. Мы подробно рассмотрим модель программирования ABAP RESTful в главах 3 и 4. Однако мы хотели бы предоставить общий обзор модели программирования ABAP RESTful в следующем разделе.

Многие основные идеи модели программирования ABAP RESTful очень напоминают модель программирования ABAP для SAP Fiori. Однако одно большое отличие заключается в том, что новая модель программирования была интегрирована в основной язык ABAP, что также делает модель доступной на SAP Cloud Platform, предоставляя базовую среду выполнения ABAP без каких-либо дополнительных инструментов и сред, таких как SAP GUI или транзакции SEGW. Кроме того, эта интеграция также обеспечивает полностью основанный на Eclipse опыт разработки E2E без необходимости доступа к транзакциям SAP GUI. Большим отличием от классической разработки ABAP в облачной среде также является использование ABAP в Git (abapGit) в качестве инструмента управления исходным кодом (SCM), заменяющего систему изменений и транспорта (CTS), обычно используемую для отслеживания изменений в коде ABAP и перемещение объектов разработки ABAP между средами ABAP. Однако в локальных установках вы по-прежнему можете использовать CTS при разработке приложений с использованием модели программирования ABAP RESTful.

#### Обзор

На рис. 1.8 показаны различные строительные блоки модели программирования ABAP RESTful. Как и в предыдущей модели программирования, в основе лежит модель данных, основанная на представлениях CDS. Если требуется только доступ для чтения к базе данных, компонент запроса берет на себя извлечение данных для службы OData. Транзакционный аналог запроса — это бизнес-объект с дополнительным определением и реализацией поведения на основе ABAP. Определение службы определяет, какие сущности CDS модели данных должны быть представлены как сущности OData. Затем привязка службы привязывает открытые сущности к протоколу связи клиента, на данный момент OData версии 2.0.

Затем бизнес-служба может быть предложена в виде пользовательского интерфейса SAP Fiori путем добавления пользовательского интерфейса элементов SAP Fiori или других клиентов пользовательского интерфейса в службу или в виде веб-API для любого потребителя, способного использовать OData REST API, например, Freestyle OpenUI5 приложения.

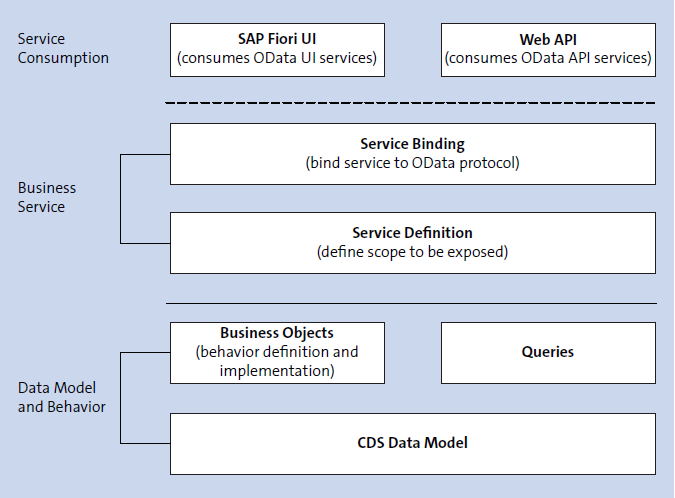


Рисунок 1.8 Архитектура модели программирования ABAP RESTful

#### Бизнес-объекты (Business Objects)

Бизнес-объект используется для обработки транзакций и определяет поведение узлов модели данных CDS (например, операции CRUD или пользовательские действия). Чтобы преобразовать модель данных CDS в дерево узлов транзакционных бизнес-объектов, отношения между узлами должны быть явно смоделированы. В предыдущей модели программирования это моделирование выполнялось с помощью аннотаций, но в модели программирования ABAP RESTful моделирование структуры бизнес-объекта было интегрировано в язык ABAP CDS. Конкретное поведение узлов в иерархии бизнес-объектов определяется с помощью нового переносимого объекта разработки ABAP, называемого определением поведения (***behavior definition***). Фактическая реализация поведения для определенного узла затем устанавливается в соответствующем классе ABAP (***implementation***).

#### Запросы (Queries)

Среда выполнения запроса берет на себя извлечение данных из базы данных для службы OData, доступной только для чтения. В отличие от поведения бизнес-объекта, для определения возможностей среды выполнения запроса не требуется никаких дополнительных артефактов. Как правило, среда выполнения запроса может преобразовывать стандартные параметры запроса OData, такие как фильтрация ($filter); пейджинг ($top, $skip); и сортировка ($orderby) в оператор SQL. Это преобразование происходит прозрачно для разработчика, что означает, что разработчику не придется иметь дело с построением оператора SQL. Другие возможности, такие как поиск, справка по значениям и предоставление текста, должны быть аннотированы разработчиком в модели данных CDS. ***Среда выполнения запроса*** обычно управляется структурой запросов (***SADL***), которая также известна из предыдущей модели программирования ABAP для SAP Fiori. SADL обеспечивает представление объектов CDS и их взаимосвязей в виде объектов OData на основе модели, которые приложения SAP Fiori затем используют без сохранения состояния через протокол OData через HTTP. Прозрачный для разработчика приложения, SADL просто связывает вместе различные уровни модели программирования без необходимости вручную кодировать доступ между различными уровнями.

Как упоминалось ранее, модель программирования ABAP RESTful будет подробно рассмотрена в главе 3.

### 1.2.4 Аналитика

Неотъемлемой частью SAP S/4HANA UX является плавная интеграция аналитического контента в стандартные этапы процесса или приложения. Бизнес-пользователи в SAP S/4HANA будут получать внутрипроцессную контекстную информацию, которая поможет в принятии решений, например о том, какие элементы обрабатывать дальше. Для пользователей, которым требуется более общее представление данных, таких как руководители или аналитики данных, SAP по-прежнему предоставляет общие инструменты ключевых пользователей для гибкого анализа бизнес-данных, например, SAP Lumira или SAP Analysis для Microsoft Excel. Сценарии встроенной аналитики для бизнес-пользователей в значительной степени поддерживаются SAP HANA, поскольку она стирает границы между классическими системами онлайн-обработки транзакций (OLTP) и онлайн-аналитической обработкой (OLAP).

SAP HANA, хранящая данные на основе столбцов и в памяти, значительно увеличила скорость агрегирования данных и позволяет создавать аналитические отчеты в режиме реального времени и обрабатывать транзакции в одной и той же системе. Еще одним средством аналитики в приложении является SAP Fiori с его богатым набором аналитических элементов управления и шаблонов, предназначенных для аналитических вариантов использования, например, страница обзора (OVP) или страница аналитического списка (ALP).

Что касается других типов приложений, то основой всего аналитического контента в SAP S/4HANA является VDM на основе представлений CDS. Аналитический контент CDS может потребляться либо через протокол OData и механизм SADL, либо через службу доступа к информации (***InA***), службы Business Intelligence Consumer Services (***BICS***), протоколы OData и аналитический механизм, как показано на рис. 1.9.

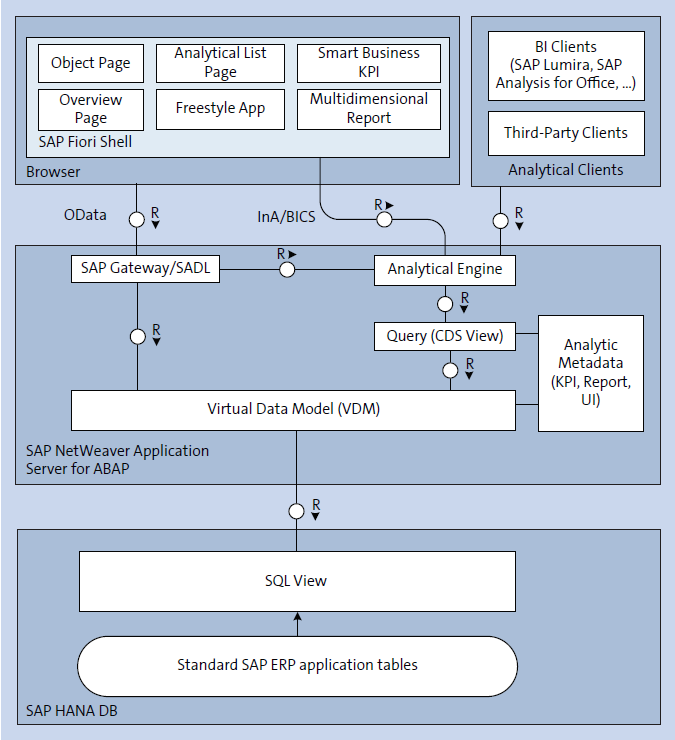


Рисунок 1.9 Аналитические части архитектуры SAP S/4HANA

Оба механизма используют аналитические аннотации CDS, например, для определения полей для агрегирования данных. Аннотация @DefaultAggregation: #SUM пометит поле представления CDS как меру, по которой данные будут агрегированы в виде суммы. Все остальные поля в представлении будут измерениями по умолчанию и формируют контекст агрегации, например, количество запросов на техническое обслуживание (мера) по типу машины (измерение). Для простых аналитических вариантов использования можно использовать механизм SADL, но для более сложных сценариев, таких как обработка иерархии или объединение исключений, необходимо использовать аналитический механизм.

Аналитическая подсистема является частью SAP BW на сервере ABAP и, как и подсистема SADL, будет использовать операторы SQL SELECT в представлениях CDS для извлечения данных. Во время выполнения не выполняется код ABAP для конкретного приложения, а выбор данных и проверки авторизации полностью передаются в базу данных SAP HANA.

## 1.3 Пользовательский опыт

Еще одна основная цель SAP S/4HANA — предложить улучшенный UX и управлять переходом от UX, характерного для традиционного программного обеспечения для бизнеса, к опыту потребительского уровня. Основой этого перехода является недавно разработанная парадигма дизайна SAP Fiori.

### 1.3.1 SAP Fiori

Пользовательский интерфейс SAP S/4HANA полностью основан на языке проектирования SAP Fiori, целью которого является предоставление облегченных приложений на основе ролей, адаптированных к конкретным задачам пользователей. Таким образом, пользовательский интерфейс SAP S/4HANA в основном основан на веб-приложениях, разработанных с помощью платформы SAPUI5, поскольку SAPUI5 является одной из поддерживаемых технологий для реализации пользовательского интерфейса SAP Fiori. SAPUI5 — это среда JavaScript, которая включает встроенную поддержку архитектурных концепций, таких как шаблон Model View Controller (MVC), двусторонняя привязка данных и маршрутизация. Приложения, разработанные с помощью SAPUI5, работают в браузерах смартфонов, планшетов и настольных ПК и адаптируются в соответствии с возможностями устройств. Подмножество SAPUI5 также стало открытым исходным кодом под лицензией Apache 2.0 под названием OpenUI5 (<https://openui5.org/>).

### 1.3.2 Панель запуска (Launchpad) SAP Fiori

Приложения SAP Fiori в SAP S/4HANA размещаются на панели запуска SAP Fiori, которая предоставляет различные услуги для приложений, например навигацию между приложениями, персонализацию и настройку приложений. Эта единая точка входа в SAP S/4HANA для конечных пользователей на любом устройстве обеспечивает доступ на основе ролей ко всем приложениям, необходимым пользователю для выполнения его конкретных бизнес-задач.

## 1.4 Выпуски SAP S/4HANA

Предложение продукта SAP S/4HANA в настоящее время состоит из двух различных выпусков: SAP S/4HANA, который является локальным выпуском (on-premise edition), и SAP S/4HANA Cloud. В следующем разделе мы рассмотрим различные архитектуры, варианты развертывания и стратегии выпуска обеих редакций SAP S/4HANA.

### 1.4.1 Локальная архитектура (On-Premise Architecture)

Объем SAP S/4HANA с точки зрения функциональности, локализации и контента можно считать эквивалентным предыдущим выпускам SAP Business Suite. Кроме того, SAP S/4HANA обеспечивает большую гибкость при использовании вашей системы. У вас по-прежнему будет полный доступ к известным инструментам администрирования (например, к транзакции PFCG для управления ролями и авторизацией), и, как и в предыдущих версиях SAP ERP, вы сможете адаптировать предварительно предоставленный контент и приложения SAP к конкретным потребностям клиентов. Кроме того, при выборе SAP S/4HANA вы можете в полной мере использовать базовую платформу SAP NetWeaver и базу данных SAP HANA для разработки пользовательских приложений SAP Fiori вместе с предварительно установленными приложениями SAP Fiori. Разработка включает полный доступ к стеку ABAP через ABAP в Eclipse, также известный как инструменты разработки ABAP. Одним из основных изменений по сравнению с предыдущими версиями SAP ERP является то, что у вас больше нет выбора поставщика поддерживаемой базы данных. SAP S/4HANA зависит от базы данных SAP HANA и создается исключительно для нее.

Вы можете установить SAP S/4HANA в своих собственных центрах обработки данных и полностью контролировать программное обеспечение и свои данные. Вы будете нести ответственность за управление всем жизненным циклом системы, включая обновления и внесение исправлений через SAP Notes. Используя этот вариант развертывания, вы получите максимальную гибкость при адаптации и настройке приложений в соответствии с вашими потребностями, а также для пользовательской разработки на платформе SAP NetWeaver, вместо того, чтобы платить лицензионный сбор за право использования программного обеспечения, что обычно требует больших затрат.

### 1.4.2 Облачная архитектура (Cloud Architecture)

По сравнению с SAP S/4HANA, SAP S/4HANA Cloud несколько более ограничен и стандартизирован. SAP предоставляет готовый к использованию контент и специализированные ключевые пользовательские инструменты для настройки, но доступ к серверной части SAP NetWeaver больше не предоставляется. Это изменение предотвращает несовместимые изменения и модификации и обеспечивает беспрепятственное ежеквартальное обновление системы. Весь доступ к системе осуществляется через браузер через панель запуска SAP Fiori, которая представляет собой оболочку, в которой размещены все приложения SAP Fiori, приложения ключевых пользователей для настройки системы и приложения конечного пользователя.

Панель запуска SAP Fiori — это единая точка входа в систему SAP S/4HANA Cloud. Если приложения конечного пользователя поддерживают настройку, их можно в определенной степени настроить и расширить с помощью инструментов ключевого пользователя. Эта возможность называется расширяемостью в приложении, поскольку расширения создаются в той же системе, что и расширенное приложение. Если в дополнение к приложениям, поставляемым SAP, требуются специальные приложения, их необходимо разработать и развернуть в сценарии параллельного расширения с использованием SAP Cloud Platform. Доступ к SAP NetWeaver AS для ABAP через средства разработки ABAP в Eclipse больше не поддерживается.

В облаке доступны два варианта подписки на SAP S/4HANA Cloud: публичное облако или частное облако. На рис. 1.10 показан общий обзор различных редакций SAP S/4HANA, в том числе локальных.

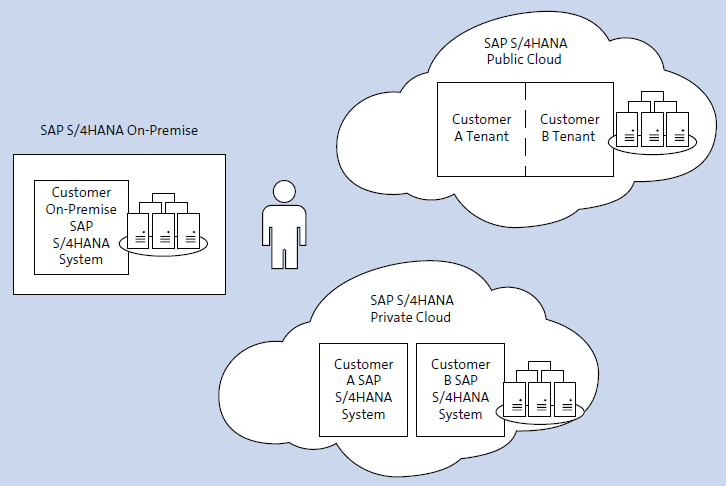


Рис. 1.10 Общий обзор различных вариантов развертывания SAP S/4HANA

#### Публичное облако(Public Cloud)

В общедоступном облаке SAP S/4HANA Cloud предоставляется SAP как программное обеспечение как услуга (SaaS) и работает в многопользовательской общедоступной облачной среде в центрах обработки данных SAP.

С точки зрения поставщика облачных услуг максимально эффективное использование доступных вычислительных и инфраструктурных ресурсов важно для снижения совокупной стоимости владения (TCO) для предоставления услуги.

Эта эффективность лучше всего достигается за счет того, что подписчики совместно используют как можно больше ресурсов, сохраняя при этом их изолированными от других подписчиков в своих собственных «арендаторах». С точки зрения подписчика изоляция от других подписчиков облачной службы необходима для сохранения конфиденциальности данных. Вы подписываетесь на услугу и постоянно платите абонентскую плату за использование услуги. Ваши ключевые пользователи и конечные пользователи могут получить доступ к системе через браузер. SAP берет на себя управление жизненным циклом системы, включая обновления системы и исправления.

#### Частное облако (Private Cloud)

В этом сценарии развертывания SAP S/4HANA Cloud развертывается в частной облачной среде, управляемой SAP, в центрах обработки данных SAP. Подписка на частный облачный сервис подразумевает более высокую степень изоляции от других подписчиков; вместо того, чтобы каждый клиент был арендатором в общей системе, каждый клиент получает свою собственную систему. Насколько далеко простираются обязанности абонента и провайдера с точки зрения управления жизненным циклом системы, зависит от индивидуальных договорных соглашений. Эти механизмы могут варьироваться от сценариев «хостинга», когда подписчик берет на себя всю ответственность за управление жизненным циклом системы, а поставщик только предоставляет систему и инфраструктуру, до полностью управляемых сценариев, когда поставщик выполняет все обновления и обслуживание. На данный момент сценарий частного управляемого облака можно реализовать с помощью SAP HANA Enterprise Cloud.

### 1.4.3 Стратегии выпуска (Release Strategies)

И SAP S/4HANA, и SAP S/4HANA Cloud поставляются с ежеквартальными обновлениями. Однако только SAP S/4HANA Cloud обеспечивает непрерывный доступ к последним инновациям, предоставляемым SAP, поскольку эта версия постоянно пополняется ежеквартальными инновационными выпусками.

С другой стороны, для SAP S/4HANA SAP предоставляет один ключевой выпуск в год, который в течение следующих трех кварталов поставляется с пакетами функций (FP), обеспечивающими бесперебойные инновации, а также исправления программного обеспечения и правовые изменения. После третьего FP выпуск SAP S/4HANA переходит в режим только обслуживания, и в конечном итоге выпускается новая версия SAP S/4HANA.

В режиме обслуживания пакеты поддержки (SP) включают в себя исправления программного обеспечения и могут включать легальные изменения для выпусков SAP S/4HANA ежеквартально. SP может также содержать усовершенствования существующих функций, например, усовершенствования, предусмотренные контрактом в программах подключения клиентов SAP, но в SP не выпускаются совершенно новые функции или инновации.

Технически FP и SP поставляются в виде стеков пакетов функций (FPS) и стеков пакетов поддержки (SPS), которые представляют собой связанные наборы функций или пакетов поддержки, которые должны применяться в данной комбинации.

На рис. 1.11 представлены различные стратегии выпуска SAP S/4HANA и SAP S/4HANA Cloud.

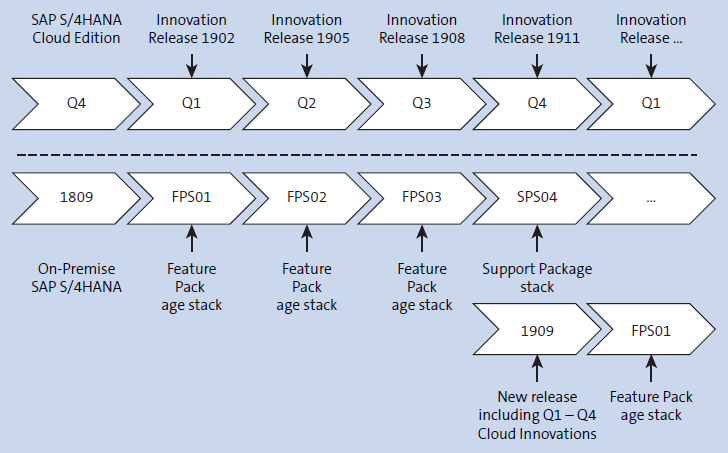


Рис. 1.11 Различные стратегии выпуска SAP S/4HANA и SAP S/4HANA Cloud

## 1.5 Облачная платформа SAP (SAP Cloud Platform)

Облачная платформа SAP играет важную роль в расширении SAP S/4HANA и SAP S/4HANA Cloud современными облачными сервисами и приложениями, как показано на рис. 1.12. SAP Cloud Platform — это предложение SAP «Платформа как услуга» (PaaS) для разработки облачных приложений в Neo (собственная среда SAP) или Cloud Foundry (облачная платформа с открытым исходным кодом, в разработке которой также участвует SAP). Кроме того, на платформе SAP Cloud Platform размещается множество инновационных сервисов и приложений SAP Leonardo в таких областях, как машинное обучение и Интернет вещей (IoT). Как клиент, подписавшийся на облачную службу SAP S/4HANA, у вас больше не будет доступа к стеку SAP NetWeaver AS для ABAP с помощью Eclipse. Вместо этого вы будете разрабатывать пользовательские приложения на SAP Cloud Platform в сценарии параллельного расширения. Клиенты SAP S/4HANA, с другой стороны, по-прежнему могут выбирать между разработкой своих пользовательских приложений в SAP NetWeaver или переходом на SAP Cloud Platform, что может облегчить миграцию в SAP S/4HANA Cloud на более позднем этапе, поскольку пользовательский код уже перенесен в среду облачной платформы и перемещен из ядра SAP ERP. SAP Cloud Platform поддерживает широкий спектр языков программирования через контейнерные среды выполнения, например Java, Node.js, ABAP и HTML5. Кроме того, SAP Cloud Platform предоставляет широкий набор вспомогательных сервисов, начиная от сервисов данных и хранения (SAP HANA, Redis на SAP Cloud Platform, PostgreSQL) и заканчивая SAP Cloud Platform DevOps (SAP Web IDE). Бизнес-сервисы и сервисы интеграции (SAP Tax Service, RabbitMQ на платформе SAP Cloud Platform, интеграция с облачной платформой SAP). Чтобы облегчить использование SAP S/4HANA VDM и API-интерфейсов OData поверх него из SAP Cloud Platform, SAP выпустила SAP Cloud SDK для Java и JavaScript/TypeScript.

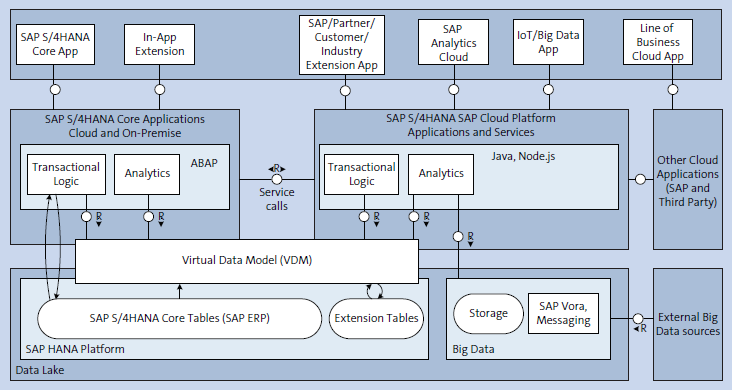


Рис. 1.12 Общая архитектура набора приложений SAP S/4HANA, включая облачную платформу SAP для разработки параллельных расширений

## 1.6 Резюме

SAP S/4HANA основан на совершенно другой архитектуре, оптимизированной для SAP HANA. Тем не менее, основная цель архитектуры SAP S/4HANA — оставаться максимально совместимой с предыдущими версиями SAP ERP, чтобы упростить миграцию и использовать существующие, хорошо зарекомендовавшие себя и проверенные бизнес-процессы, разработанные SAP, партнерами и клиентами в последние десятилетия. Краеугольным камнем SAP S/4HANA является VDM, основанный на CDS, который в значительной степени обеспечивает перенос кода в SAP HANA в соответствии с парадигмой Code-to-Data. VDM — это основа всех типов приложений в SAP S/4HANA, от транзакционных и аналитических до внешних интерфейсов и ESH. Явно выпущенные части VDM также могут использоваться партнерами и клиентами в качестве основы для разработки пользовательских приложений. В SAP S/4HANA 1909 (выпущенном в сентябре 2019 г.) модель программирования ABAP RESTful теперь является рекомендуемой моделью программирования для разработки транзакционных приложений SAP Fiori в SAP S/4HANA и заменяет модель программирования ABAP для SAP Fiori. Обновленный UX на базе SAP Fiori — еще один большой шаг вперед, и он будет подробно рассмотрен в следующей главе.