



Ivan Barac

Иван Барач

Эрикссон Никола Тесла а.о., Загреб, Хорватия
Ericsson Nikola Tesla d.d., Zagreb, Croatia

Ключевые слова:

**Платформа для поставки услуг, SDP
Окружение для создания услуг, SCE
Ориентированная на услуги архитектура, SOA**

Услуги Web

Поставщик услуг

Поставщик содержаний

Parlay/OSA

Parlay X

Конвергенция

Key words:

Service Delivery Platform, SDP

Service Creation Environment, SCE

Service Oriented Architecture, SOA

Web Services

Service provider

Content provider

Parlay/OSA

Parlay X

Convergence

Окружение для создания услуг

Резюме

Окружение для создания услуг (SCE – Service Creation Environment) это набор компонентов, обеспечивающих оператору сети возможность поставки новых цифровых услуг посредством горизонтально построенной сети услуг. Основной задачей платформы для поставки услуг (SDP – Service Delivery Platform) является поддержка и интеграция деловых и эксплуатационных процессов оператора. Окружение для создания услуг поддерживает полный жизненный цикл услуг: замысел, выполнение, активирование, использование, оплату и управление услугами и их изъятие из использования.

Окружение для создания услуг конструировано для поставки услуг посредством сетей с множественным доступом. В традиционных сетях окружение для создания новых услуг позволяет оператору или самостоятельное создание услуг, или в тесном сотрудничестве с несколькими партнерами, т.н. “walled garden” (обнесенный стеной сад) деловая модель. Однако операторы начали рассматривать и искать решения, которые бы им позволили партнерские отношения с более широкой группой поставщиков приложений/решений. Та-

кой способ работы требует от интерфейсов взаимодействия B2B “бизнес для бизнеса” (Business to Business) поддержку для установления отношений между операторами и поставщиками услуг, а также совместный доступ к сетевым прикладным услугам, разработанным партнерами.

Для быстрого и рентабельного введения новых услуг на рынок, интерфейсы B2B должны обеспечивать партнерам простоту в работе, а управление процессами взаимодействия оператора с большим числом партнеров должно быть автоматизировано.

Окружение для создания услуг это место встречи мира информатики с миром телекоммуникаций. Некоторые новые технологии особенно пригодны для этой цели, прежде всего это относится на серверы приложений для разработки и внедрения приложений, а также web услуги (Web Services), как технология для интеграции интерфейса B2B и услуг внутри домена оператора. Платформа компании Эрикссон для поставки услуг базируется на упомянутых технологиях и обеспечивает архитектуру, поддерживающую оператора в его стремлениях к использованию деловых моделей сотрудничества с большим числом партнеров. Это решение предоставляет совместное окружение для разработки и внедрения пользовательских услуг, обеспечивая доступ к сетевым ресурсам собственным и внешним разработчикам приложений и поставщикам услуг.

Service Creation Environment

Abstract

A Service Creation Environment (SCE) is a set of components that allows a network operator to deliver new digital services through a horizontal Service Network. A fundamental requirement is that the Service Delivery Platform (SDP) supports and tightly integrates with the operator's business and operational processes.

The Service Creation Environment supports complete service life-cycle management, including creation, deployment, provisioning, utilization and withdrawal, charging as well as operation and maintenance.

The Service Creation Environment is designed to deliver services over multiple access networks. Traditionally, the operators' environment for developing and launching new end-user services supported service creation by either in-house personnel or in close cooperation with several partners (“walled garden” business model). However, operators are now looking for solutions that will enable them to partner with a much larger group of application developers/providers. This puts new requirements on the Business to Business (B2B) interfaces used with regard to business cooperation of operators and their business partners, in particular with regard to providing access to the Service Creation Environment both to operators and their partners. To reduce the time, cost and complexity in bringing new end-user services to the market, the B2B interfaces must be easy-to-use for partners and the process of handling many partners must to a large degree be automated for the operator. The service creation environment is a meeting place for the IT and telecom worlds. Some important emerging technology winners here are application servers providing the development and deployment infrastructure for applications, and web services as the technology for B2B interfaces.

Ericsson's service delivery platform utilizes the above technologies and provides an architecture that supports the operator when growing into business models supporting mass partnering/mass application scenarios. The solution provides a common development and execution environment for end-user services, facilitating access to network capabilities for both internal and external application developers and service providers.

1. Введение

Окружение для создания услуг компании Эрикссон (SCE – Service Creation Environment) и платформа для поставки услуг (SDP – Service Delivery Platform) представляют единственное решение для сетей услуг, позволяющее полную конвергенцию услуг, независимо от доступа и типа пользовательского терминала. В склонном переменах мире коммуникаций, учитывая потребности пользователей и в соответствии с предвидением компании Эрикссон о неизбежности коммуникаций, платформа для поставки услуг своей открытостью, архитектурой, конструкцией и возможностями в целостности выполняет ожидания и требования сетевых операторов, поставщиков услуг и содержаний, и конечных пользователей этих услуг. Осуществление такой комплексной задачи невозможно без целостного подхода к решениям коммуникации, а именно такой подход характеризует технологии и традицию компании Эрикссон.

Потребность обеспечения разработчикам приложений и поставщикам услуг доступа к возможностям сети становится решающим элементом для увеличения сетевой нагрузки и прибыли оператора. Платформа для поставки услуг, удовлетворяющая запросам, связанным с быстрым развитием и введением новых услуг, ключевой фактор успеха этого намерения.

Цель этой статьи познакомить читателей со сложностью и различными потребностями этой платформы, но и с остальными протагонистами современного делового окружения. Это относится не только на информационные и коммуникационные технологии, но также и на другие деятельности, а, прежде всего, на средства информации. Во второй главе описаны деловые процессы и возможные сценарии взаимодействия между заинтересованными сторонами (оператор, разработчик приложений, поставщик содержаний, поставщик услуг..).

Третья глава предлагает обзор функциональных запросов и логических архитектур окружения для развития и поставки услуг.

В четвертой главе описаны принципы, образцы, архитектуры и технологии, используемые в окружении для создания услуг.

2. Деловые модели и функции отдельных участников в создании услуг

2.1. Деловые функции

Протагонисты, включенные в создание услуг для конечного пользователя, могут появляться в разных ролях, т.е. функциях. Эти функции являются основой описания современных деловых моделей, перечисленных в главе 2.2 этой статьи.

На Рис. 1. представлены главные деловые функции в окружении для создания услуг и отношения между ними.

2.2. Описание деловых функций в окружении для создания услуг

2.2.1. Конечный пользователь

Конечный пользователь (EU – End User) является пользователем услуг, приложений и сетевого доступа. Конечным пользователем может быть организация или индивидуальный пользователь. Особым случаем

оконечного пользователя является приложение, которое, например, использует сетевую услугу позиционирования для определения местоположения пользователя.

В отношениях между индивидуальным конечным пользователям и собирателем/агрегатором содержаний и приложений (CAA – Content and Application Aggregator) очень большое значение имеет доверие, защита конфиденциальности и защита от нежелательных содержаний (spam- мусор). Для выполнения таких условий, важно обеспечить конечному пользователю поддержку для активного использования услуги, на которую пользователь подписан.

2.2.2. Оператор сети

Следует подчеркнуть, что оператор сети электросвязи (NP – Network Provider) в реальности представлен группой деловых функций: поставщика доступа, поставщика оплачиваемых услуг, поставщика идентификационных характеристик (access provider, raument provider, identity provider), и т.п. Однако так как на рынке оператор почти всегда появляется во всех этих функциях, из-за простоты он здесь описан как протагонист одной роли: оператора сети.

Оператор сети обеспечивает:

1. конечным пользователям доступ к ряду коммуникационных услуг;
2. поставщикам приложений (Application Providers) различные сетевые прикладные услуги, такие как, например, определения местоположения, присутствие пользователя, контроль вызова и т.п.;
3. специфические компоненты и услуги собирателям/агрегаторам содержаний и приложений, обеспечивая им возможность более простого предложения содержаний и приложений конечным пользователям. Примером таких услуг являются идентификационные характеристики пользователя и тарификация.

Можно сказать, что двумя самыми значительными ценностями, связанными с оператором сети, являются инфраструктура коммуникаций и качественные отношения со всеми конечными пользователями сети.

2.2.1. Собиратель/агрегатор содержаний и приложений

Собиратель содержаний и приложений обеспечивает конечному пользователю доступ к многочисленным услугам с дополнительной стоимостью (VAS – Value Added Service), таким как, например, содержания (Streaming, сохранение содержания), интерактивные услуги (игры, присутствие), персональные услуги (банковские дела), коммуникация (e-mail), m-коммерция по сотовой связи (m-commerce), и т.д.

В большинстве случаев это не единственная роль этого протагониста. Можно комбинировать эту роль с ролью оператора сети, или с ролью поставщика приложений.

2.2.2. Поставщик приложений

Поставщик приложений (AP – Application Provider) служит для управления приложениями, предлагаемыми конечным пользователям, а также для продажи разрешений на использование приложений на коммерческих принципах.

2.2.3. Разработчик приложений

Разработчик приложений (AD – Application Developer) это протагонист, который разрабатывает и тестирует приложения, а также занимается продажей лицензий на использование приложений. Разработчик приложений, чаще всего, это малый или независимый средний

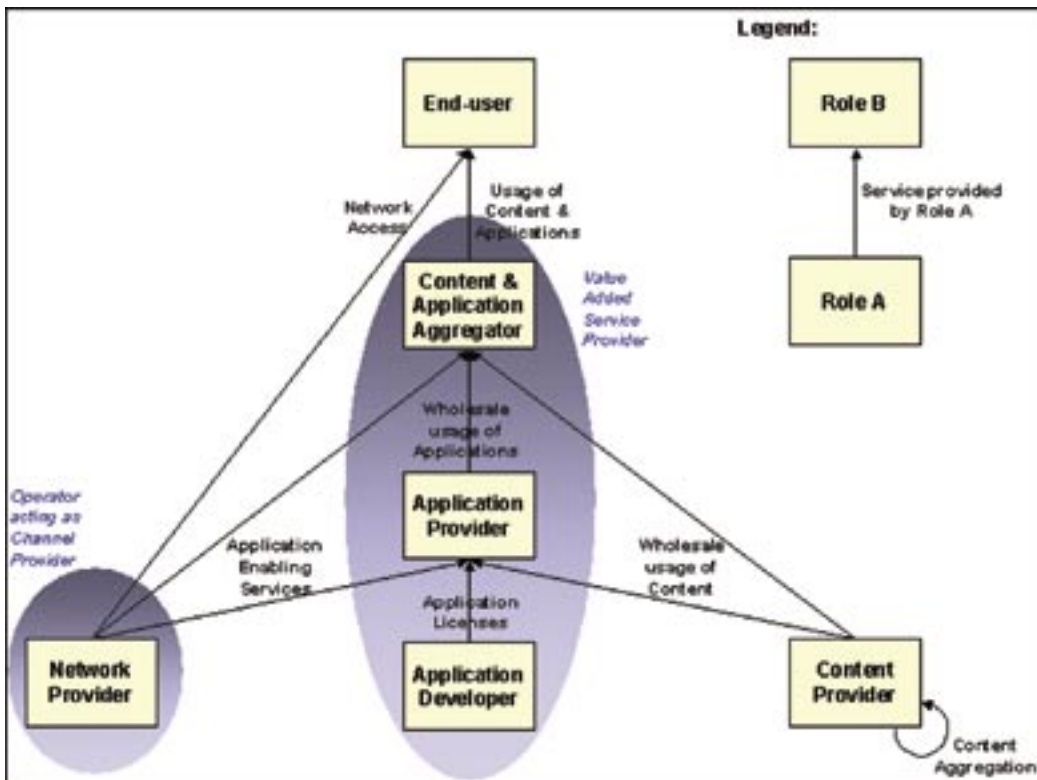


Рис. 1. Главные деловые функции в окружении для создания услуг

поставщик программного обеспечения (ISV - Independent Software Vendor), но это могут быть и большие компании, и отдельные лица.

Важно подчеркнуть следующий факт, в процессе разработки приложения сам разработчик приложения обязан временно играть роль поставщика приложений, как показано на рис. 2. Из этого следует вывод, что оператору сети требуется сотрудничество с разработчиками приложений для обеспечения простого включения прикладных сетевых услуг в процесс развития приложений для окончательного пользователя.

Чтобы оператор мог сотрудничать с большим числом разработчиков приложений, он должен принять во внимание несколько предварительных условий.

Например, большинство разработчиков приложений:

- обладают малыми знаниями о специализированных технологиях, используемых для развития мобильных приложений;
- располагают ограниченными возможностями и их не привлекают длительные переговоры об установлении отношений с операторами;
- требуют помощи при верификации приложений в окружении оператора.

2.3. Деловые модели и протагонисты

На рынке различные протагонисты выступают в одной или больше ролей, описанных в предыдущей главе, в зависимости от применяемой деловой модели (Business Models and Actors – Деловые модели и актеры). Важно подчеркнуть, что деловые различные модели взаимно не исключаются. Наоборот, в большинстве случаев несколько деловых моделей используются одновременно, в зависимости от потребностей самой услуги, предлагаемой окончательному пользователю.

Некоторые из деловых моделей, базированных на ролях, определенных в главе 2.2, описываются в продолжение статьи. Роли или функции описаны с точки зрения покупателя платформы для поставки услуг, т.е. оператора сети.

Legend	Легенда
Role B	Функция B
Role A	Функция A
Service provided by Role A	Услуга поставленная функцией A
End-user	Оконечный пользователь
Network access	Сетевой доступ
Usage of Content & Applications	Использование содержаний и приложений
Value Added Service Provider	Поставщик услуг с дополнительной стоимостью
Content & Application Aggregator	Собиратель содержаний и приложений
Wholesale usage of Applications	Оптовая продажа использования приложений
Application Provider	Поставщик приложения
Application Licenses	Лицензии на приложение
Application Developer	Разработчик приложения
Operator acting as Channel Provider	Оператор действует как поставщик канала
Application Enabling Services	Услуги позволяющие приложения
Network Provider	Оператор сети
Wholesale usage of Content	Оптовая продажа использования содержаний
Content Provider	Поставщик содержания
Content Aggregation	Собиратель содержаний

Оператор сети обеспечивает сетевую инфраструктуру, однако он может выступать и в некоторых других ролях.

2.3.1. Оператор как традиционный поставщик услуг

Ранее операторы связи были единственными поставщиками услуг окончательным пользователям – Оператор как традиционный поставщик услуг: историческая модель (Operator as Traditional Service Provider: The Historical Model). Точнее, кроме обеспечения сетевой инфраструктуры, оператор играл роль собирателя содержаний и приложений, поставщика приложений, а часто и содержаний, а также разработчика приложений.

Ниже перечислены преимущества оператора:

- внутренние (неформальные) отношения при предоставлении сетевых услуг и услуг поддержки, которые облегчают обеспечение надежности, оплату услуг и т.п.;
- чистая прибыль от продажи содержаний и приложений остается оператору.

Однако в существующей модели невозможно остаться

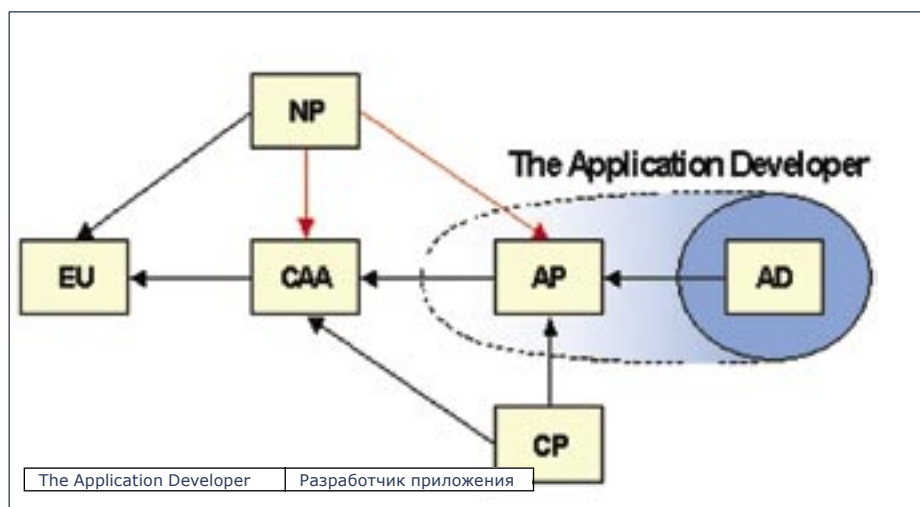


Рис. 2. Позиция разработчика приложений

поставщиком всех услуг, требуемых окончательным пользователям, и при сравнении с конкуренцией это является ясным недостатком.

2.3.2. Оператор как поставщик доступа

Одной из укоренившихся деловых моделей в настоящее время является модель, в которой оператор выступает в функции поставщика доступа окончательным пользователям, без предоставления сетевых услуг или услуг поддержки внешним партнерам - Оператор как "чистый" поставщик доступа: Модель услуг сети Интернет (Operator as Pure Access Provider: The Internet Services Model). Типичным примером такой модели являются услуги, которые пользователи находят на Интернете.

В этом случае преимуществом оператора является тот факт, что окончательный пользователь, подключаясь к содержимым и приложениям в сети Интернет, увеличивает нагрузку, а значит и прибыль поставщика доступа.

Недостатки этой модели следующие:

- оператор не является частью финансовой цепи приложений и содержаний (нет разделения прибыли);
- приложения обладают ограниченными функциональными возможностями, т.к. не могут использовать сетевые услуги;
- потенциалы оператора в отношении поставки сетевой услуги и/или услуги поддержки не использованы.

2.3.3. Оператор как поставщик услуг

В настоящее время некоторые значительные в мире операторы поддерживают деловую модель, в которой они способствуют в применении услуг окончательными пользователями, а их деловые партнеры разрабатывают и обслуживают приложения, которые обеспечивают предлагаемые услуги - Оператор как поставщик услуг: Модель маркированных услуг (Operator as Service Provider: The Branded Services Model).

В последнее время операторы используют т.н. модель "обнесенного стеной сада" (Walled garden), в которой, наряду с договором об уровне услуги (SLA - Service Level Agreement), с малым числом поставщиков приложений заключается договор о совместном маркетинге услуг.

Однако существуют комплексные технические и административные барьеры, часто формализованные процессом квалификации, что препятствует установлению партнерских отношений. Другими словами, услуги, предлагаемые посредством интерфейса, обозначенного красной стрелочкой на рис. 5., нынче недоступны большинству деловых потенциальных партнеров.

Преимущества оператора, как поставщика услуг, следующие:

- оператор контролирует финансовую цепь содержаний и приложений;

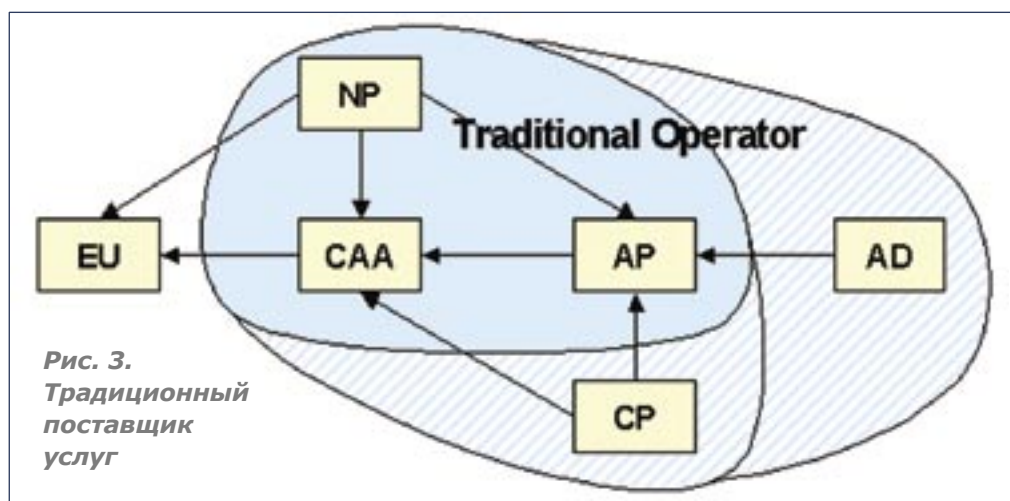


Рис. 3. Традиционный поставщик услуг

Traditional Operator

Традиционный оператор

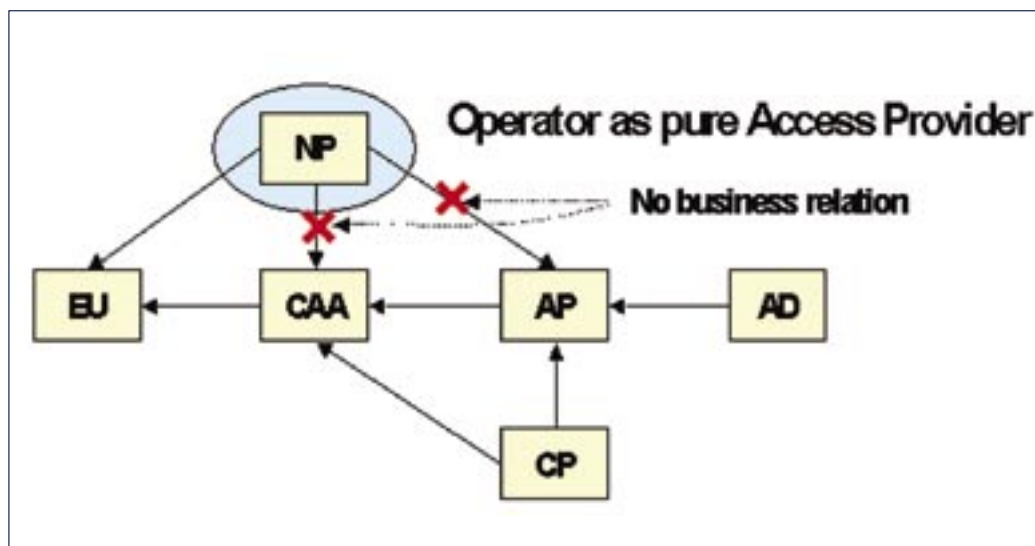


Рис. 4. Оператор как "чистый" поставщик доступа

- оператор располагает возможностью предоставления сетевых услуг внешним партнерам для использования в их приложениях.

Operator as pure Access Provider	Оператор как "чистый" поставщик доступа
No business relation	Нет деловых отношений

Недостатки этой модели:

- сетевые услуги, предоставляемые партнерам, ограниченные и неудобные, и поэтому партнеры не могут их достаточно эффективно использовать;
- деловые интерфейсы в направлении поставщиков приложений построены по образцу "малое число поставщиков = большие, хорошо известные поставщики", что исключает большое число поставщиков приложений, а этим уменьшается число услуг, которые могут быть предложены конечным пользователям.

2.3.1. Оператор как поставщик каналов

В последние годы операторы все чаще выступают в роли поставщиков каналов – Оператор как поставщик каналов: Модель полной услуги (Operator as Channel Provider: The Full Services Model). Это значит, что оператор предоставляет большому числу поставщиков услуг не только возможность использования ресурсов

сети, таких как, например, управление вызовом, присутствие, определение местоположения конечного пользователя, и т.п. Оператор также предоставляет им и услуги поддержки, связанные с управлением идентификационными характеристиками (Identity Management): проверка подлинности, проверка полномочий, управление системой тарификации, и т.п. Деловые интерфейсы, обозначенные красной стрелкой на рис. 6., таким образом, становятся еще более важными.

Ниже перечислены преимущества оператора:

- оператор обладает большей частью финансовой цепи содержаний и приложений, что ему обеспечивает увеличение прибыли;
- оператор может своим деловым партнерам предоставить сетевые услуги для использования в их приложениях ("a network in every application" – сеть в каждом приложении);
- оператор может своим деловым партнерам обеспечить услуги поддержки.

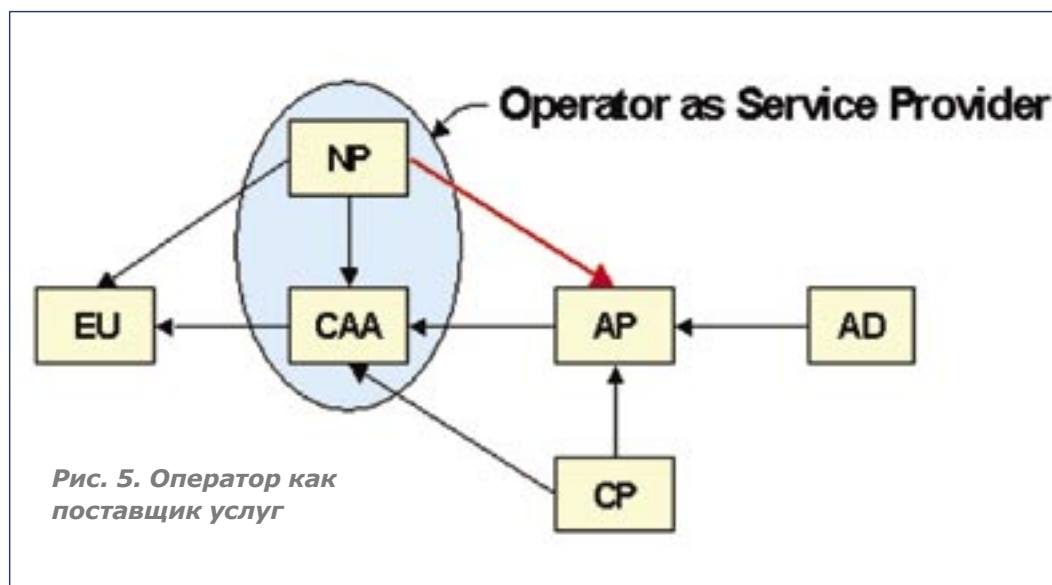


Рис. 5. Оператор как поставщик услуг

Operator as Service Provider	Оператор как поставщик услуг
------------------------------	------------------------------

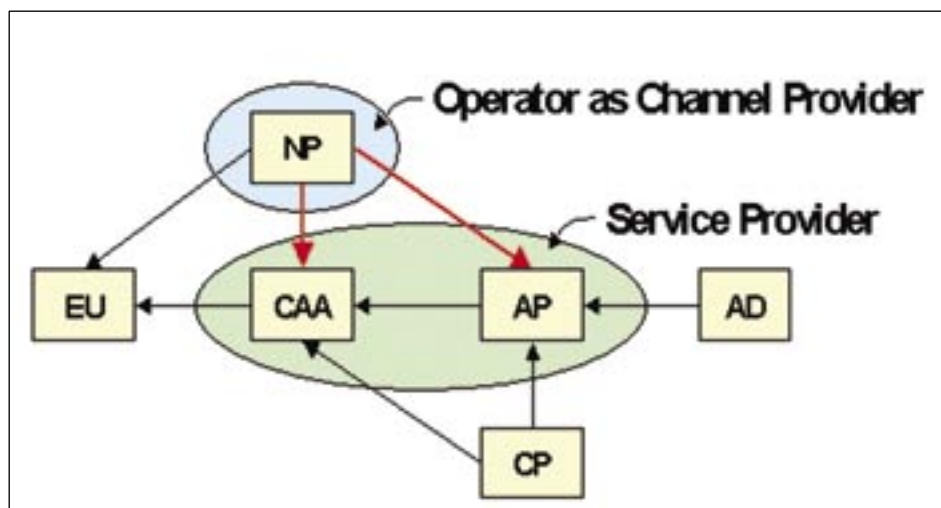


Рис. 6. Оператор как поставщик каналов

Недостатком этой модели является ограниченная прибыль от содержаний и приложений по сравнению с моделью, в которой оператор их предлагает самостоятельно.

2.4. Гибкая поддержка деловых моделей

Как уже упомянуто, операторы не используют только одну из возможных моделей, они стараются выбрать самую лучшую модель для каждой специфической услуги и специфической группы конечных пользователей. Одним из самых больших препятствий на пути к желаемой гибкости является потребность комплексной и требующей много усилий интеграции,

Operator as Channel Provider	Оператор как поставщик каналов
Service provider	Поставщик услуг

специфичной для промышленности телекоммуникаций. Поэтому задаваемая цель может быть достигнута только с помощью платформы для поставки услуг, которая сможет поддерживать и будущие потребности (future-proof platform – перспективная платформа).

2.4.1. Преимущества оператора

Такая платформа обеспечивает операторам множество функциональных ключевых преимуществ, которые укрепляют его деятельность. В продолжении перечислены преимущества не только для сетевых

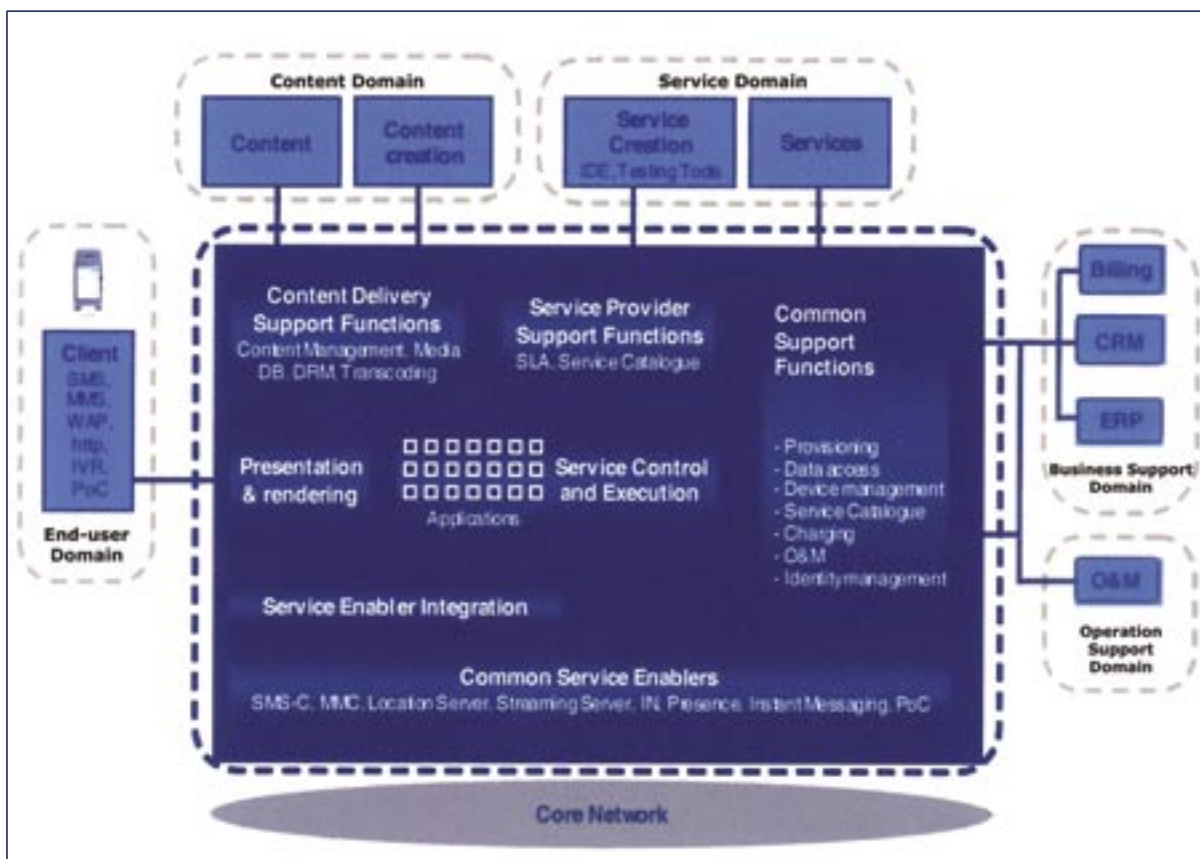


Рис. 7. Платформа для поставки услуг – логическое изображение

операторов, но также и для собирателей приложений и содержаний, и для поставщиков приложений, имея в виду то, что оператор, в большей или меньшей степени, также будет действовать в каждой из этих ролей:

- сокращается время поставки услуг на рынок (shorter TTM, shorter Time to Market), от замысла услуги до поставки посредством использования коммерческих серверов приложений, инструментов для развития, и доступ широкому кругу разработчиков приложений;
- оптимальная оплата услуг конечным пользователям, с использованием гибкой модели оплаты, например, оплата каждого события, абонентская подписка, оплата содержания, и т.п.;
- статус каждой услуги, предлагаемой конечному пользователю, известен в каждый момент времени (service assurance – гарантия услуги);
- большое число поставщиков услуг, с которыми устанавливаются отношения посредством автоматизированных процессов и интерфейсов, например, договор об уровне услуги;
- окружение, в котором объединено множество различных поставщиков (multivendor environment – неоднородная среда), управляемое архитектурой, которая поддерживает простую адаптацию разных интерфейсов.

2.4.1. Преимущества разработчиков приложений

- сокращается время включения сетевых услуг, что обеспечено простыми интерфейсами и доступом, а также стандартными инструментами для разработки;
- простой доступ к сетевым услугам оператора, без комплексных договоров.

2.4.2. Преимущества окончного пользователя

- простой доступ, использование и управление большим числом услуг, независимо от категории и типа содержаний и приложений;
- рост числа и спектра услуг, благодаря экономически эффективной разработке, внедрению и выполнению новых приложений;
- выгодные цены, вследствие конкуренции на рынке поставщиков услуг.

3. Концепт платформы для поставки услуг

Концептуальная логическая архитектура платформы для поставки услуг (SDP - Service Delivery Platform) представлена на рис. 7. Можно заметить, что сеть услуг разделена на целый ряд логических функций, составляющих единственную целостность, которые в состоянии выполнить деловые запросы оператора, играющего роль поставщика каналов в течение жизненного цикла услуги.

Платформа для поставки услуг это целостное решение, которое поддерживает полный жизненный цикл услуги, и может быть разделено на три концептуальные целостности, содержащие требуемые им функции:

- Окружение для создания и поставки услуг
- функции поддержки содержаниям
- функции поддержки поставщикам услуг
- презентация и изображение содержаний приложения
- Общие функции поддержки
- активирование услуг
- доступ к данным
- каталог услуг

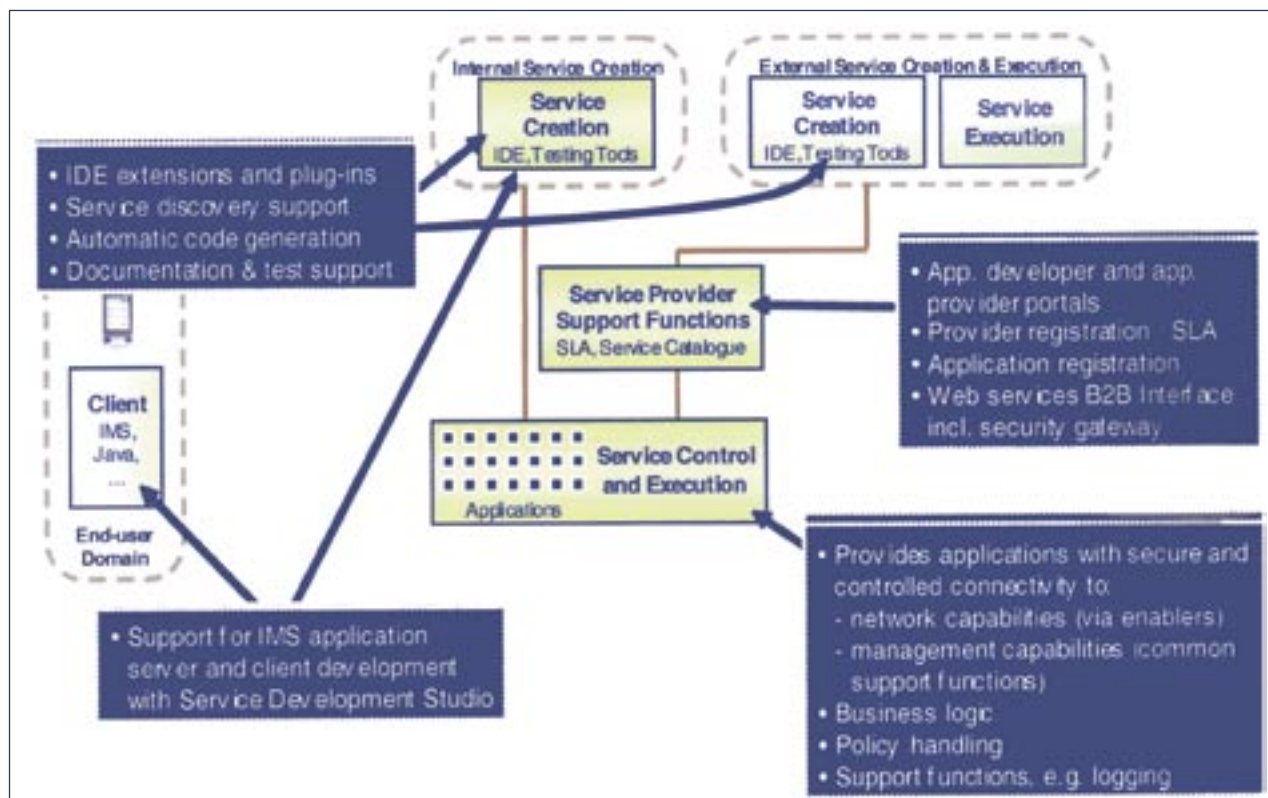


Рис 8. Окружение для создания и поставки услуг

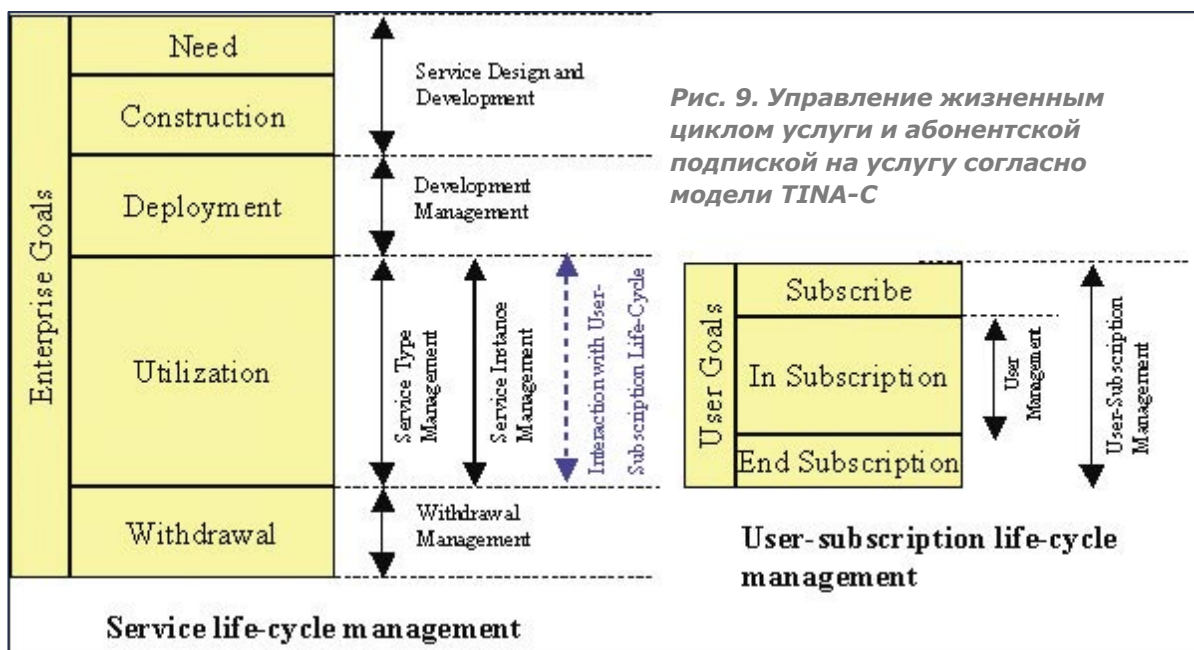


Рис. 9. Управление жизненным циклом услуги и абонентской подпиской на услугу согласно модели TINA-C

Enterprise Goals	Цели предприятия
Need	Потребность
Construction	Конструкция
Deployment	Внедрение
Utilization	Использование
Withdrawal	Изъятие из эксплуатации
Service Design and Development	Замысел и разработка услуги
Development Management	Управление разработкой
Service Type Management	Управление типом услуги
Service Instance Management	Управление экземпляром услуги
Interaction with User – Subscription Life-Cycle	Взаимодействие с пользователем – подписка на жизненный цикл
Service life-cycle management	Управление жизненным циклом услуги
User Goals	Цели пользователя
Subscribe	Подписка
In Subscription	Подписка активна
End Subscription	Конец подписки
User Management	Управление пользователем
User-Subscription Management	Управление подпиской пользователя
User-Subscription life-cycle management	Управление жизненным циклом подписки пользователя

поставки услуг, а указаны и влияния различных аспектов на окружение.

4. Создание и поставка услуг

4.1. Концепт жизненного цикла услуги

При создании услуг следует иметь в виду следующие фазы жизненного цикла услуги:

- Разработка (development)
- Внедрение (deployment)
- Использование (utilization)
- Изъятие из эксплуатации (withdrawal).

Эта терминология согласована с TINA-C моделью услуг (Telecommunications Information Networking Architecture Consortium). На рис. 9 представлен жизненный цикл услуги и абонентской подписки на услугу.

На рис. 9 представлена грубая модель TINA-C, которую можно разработать детальнее, что сделано на Рис. 10.

Терминология, использованная в детальной разработке жизненного цикла услуги, может быть отображена eTOM процессами, как следует в продолжение текста:

- оплата услуг
- обслуживание и управление услугами
- управление идентификационными характеристиками пользователей и партнеров
- Общий слой интеграции
- OSA/Parlay шлюз
- SMS, служба коротких сообщений
- MMS, служба мультимедийных сообщений
- сервер для мгновенных сообщений
- поточный (streaming) сервер
- сервер для мобильного позиционирования
- сервер присутствия.

Детальное изображение окружения для создания и поставки услуг находится на Рис. 8., с акцентом на технологические и деловые решения этой части целостной платформы для поставки услуг. В продолжение статьи описано окружение для создания и

- Управление жизненным циклом услуги (Service Life Cycle Management):
- разработка (development) является частью eTOM (enhanced Telecom Operations Map) управления жизненным циклом услуги (Product Life Cycle Management)
- внедрение и изъятие из использования (Deployment &Withdrawal) являются частью eTOM процесса поддержки и готовности операций - Operations Support &Readiness
- использование (utilization) является частью eTOM процесса выполнения, обеспечения и оплаты услуги - Fulfillment, Assurance and Billing.
- Управление жизненным циклом абонентской подписки на услугу (user-subscription life cycle management)
- подписка (subscribe) является частью eTOM процесса выполнения - Fulfillment
- подписка активна (in subscription) является частью eTOM процесса обеспечения и оплаты услуги - Assurance and Billing

4.2. Окружение для создания услуг и технологические домены

Конвергенция телекоммуникаций (Telecom) с коммуникациями данными (Datacom) рассматривается уже много лет. На Рис. 12. представлен результат одной из таких попыток (Yankee Group).

В этом концепте конвергенция описана как конфликт. И это до определенной меры истина. Но вопреки этому, разные производители телекоммуникационного оборудования (Telecom) и оборудования для коммуникации данными (Datacom) видят выход из этого конфликта в решениях, которые объединяют преимущества каждой из областей и, поэтому, более эффективные по сравнению с предыдущими специфическими решениями для отдельных доменов.

Решение компании Эрикссон представлено на Рис. 13. Это решение в целом использует технологии и инфраструктуру, которую могут обеспечить Datacom компании с целью использования Telecom специфических возможностей в информационных и коммуникационных приложениях с целью предоставления услуг окончному пользователю.

Окружение для создания услуг это программное решение, которое применяет современные концепты, принципы, образцы и самые лучшие искусства. В продолжение статьи дано объяснение, инструкции и примеры такого общего и межпромышленного подхода и технологии.

Service Life Cycle	Жизненный цикл услуги
Development	Разработка
Deployment	Внедрение
Design	Конструкция
Develop	Развитие
Register	Регистрация
Install	Установка
Deploy	Развертка
Provide	Поставка
Idea	Идея
Track Behaviour	Наблюдение за поведением
Withdraw	Изъятие
Manage	Управление
Charge	Оплата
Access	Доступ
Activate	Активирование
Discover	Опознавание
Service Availability	Доступность услуги
Campaign	Кампания
Package	Пакет
Offer	Предложение
Application Developer	Разработчик услуги
Service Provider	Поставщик услуги
End-User Service	Услуга окончного пользователя
End-User	Оконечный пользователь

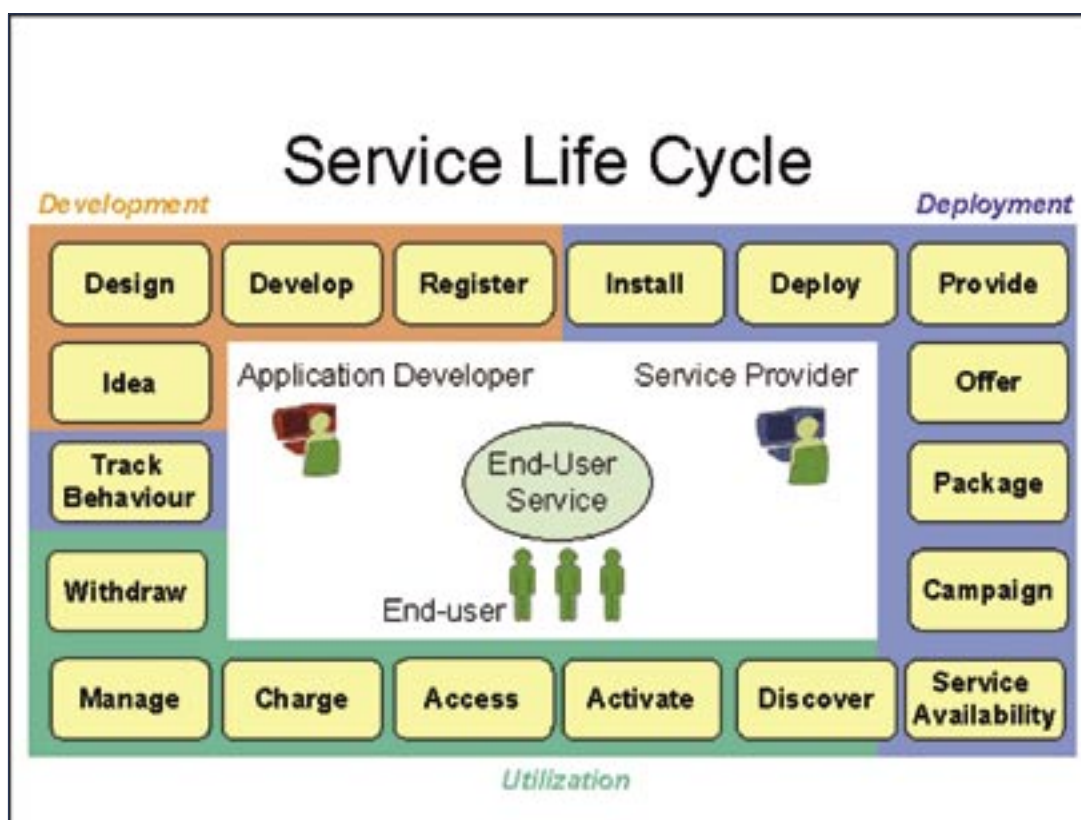


Рис. 10. Детальная разработка жизненного цикла услуги

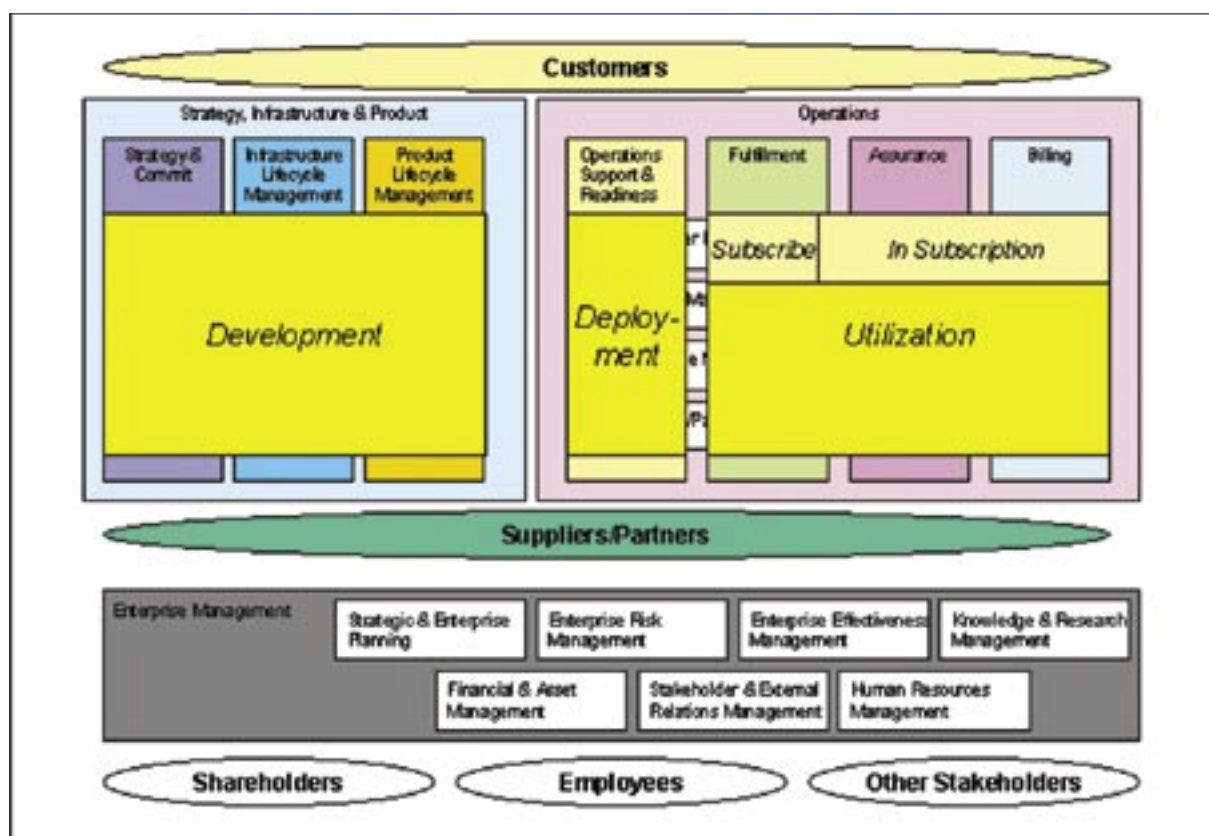


Рис. 11. Управление жизненным циклом услуги и абонентской подписки на услугу и eTOM

4.2.1. Распределенная многослойная архитектура прикладной поддержки

Многослойный подход это способ структурирования приложений (прикладного уровня). Такой способ, например, используется для специфических частей под названием front-end/presentation/facade (клиентская часть/презентация/фасадный метод), частей деловой логики под названием generic/core/reusable (общие/базовые/с повторным использованием), и специфических частей back-end/data/resource connection (серверная часть/данные/ресурсы), как это представлено на Рис. 14. Распределенная многослойная архитектура программной поддержки хорошо известный принцип и применяется, до определенной меры, в каждом приложении программной поддержки.

Этот концепт базируется на разделении понятий, (separation of concern), образце, который поддерживает более простое обслуживание и наращивание программной поддержки, одновременно способствуя идее повторного использования (re-use) отдельных частей программной поддержки.

Многослойный подход может быть применен и в логической, и в физической части решения. Также важно заметить, что многослойная структура в целом может быть сформирована каскадно, например, в случае, когда определенная прикладная система является частью другой, большей прикладной системы.

Компоненты каждого слоя, а особенно слоя базовой логики (core logic), можно также складывать каскадами внутри слоя. Основная структура каждого слоя может быть применена рекурсивно на разные уровни системы или компоненты, от самой высшей логической части до низших логических частей системы, а также включая и

Customers	Заказчики
Suppliers/Partners	Поставщики/Партнеры
Shareholders	Акционеры
Employees	Служащие
Other Stakeholders	Другие организаторы совместной деятельности

самые малые компоненты, при условии, если решение структурировано по принципу – представительский слой фасада, слой базовой логики и слой соединителя ресурсов.

Названия слоев могут зависеть от словаря, применяемого в различных доменах или специфических реализациях отдельных технологических доменов. Однако подход, который поддерживает принципы и содержит компоненты базовой логики, адаптации/посредничества в направлении клиентов (front-end) и адаптации/посредничества в направлении ресурсов (back-end), аналогичный для всех типов систем прикладной поддержки (software systems).

Запросы, которые могут быть заданы к каждому слою, следующие:

- фасад представления (Presentation Façade): компоненты представляют услуги посредством web портала (приспособлено различным пользовательским сегментам), компоненты представляют услуги как web услуги (приспособленные различным стандартам), надежность, и т.д.;
- базовая логика (Core Logic): компоненты предоставляют деловые базовые услуги на различных уровнях, (компонент базовой логики реализует деловой процесс или часть процесса, и использует другие компоненты базовой логики и/или соединителя ресурсов);
- соединитель ресурсов (Resource Connector): компоненты предоставляют (синтаксическую) адаптацию и (семантическое) посредничество в направлении ресурсов.

Спецификация прикладной платформы J2EE (Java 2 Enterprise Edition – технология разработки корпоративных приложений на языке Java, вариант 2) является примером спецификации, поддерживающей распределенную многослойную архитектуру прикладной поддержки (software architecture). Распределенная структура J2EE состоит из следующего:

- компонентов клиентского слоя (Client-tier), которые выполняются на оборудовании клиента;
- компонентов web слоя (Web-tier), которые выполняются на J2EE прикладном сервере;
- компонентов делового слоя (Business(EJB)-tier), которые выполняются на J2EE прикладном сервере;
- слоя корпоративных информационных систем (EnterpriseInformation system (EIS)-tier), который выполняется на EIS прикладных серверах.

SNF рекомендация компании Эрикссон, (Ericsson SNF Recommendations), внутри технологической области – XML Web услуги, второй пример, для которого можно идентифицировать четыре логических слоя:

- слой делового фасада (Business Façade tier)
- слой деловой основы (Business Core tier)
- слой деловых ресурсов (Business Resource tier)
- слой инфраструктурных услуг (Infrastructure Services tier); в действительности это категория общих ресурсов.

4.2.2. Многоуровневая архитектура прикладной поддержки

Многоуровневый подход это способ структурирования прикладной поддержки на специфические части приложения и общие части - инфраструктурные части/платформы, где платформа/инфраструктурная часть может состоять из нескольких уровней.

Чтобы реализация и управление приложениями были эффективными, важно выбрать платформу, которая поддерживает простую интеграцию между слоями, и которая может быть использована для всех слоев целевого домена системы. Такие семейства платформ обеспечивают окружение, которое обычно основывается на известной модели компонентов, например, J2EE или .NET.

Важно заметить, что разделение системы прикладной поддержки на различные уровни относится на реализацию интерфейсов протоколов и частей прикладной поддержки.

Требуемые функции каждого уровня:

- поддержка представительского фасада (Presentation Façade Support): порталы/графические интерфейсы, представление на различных каналах и устройствах, web

Yankee Group, Wireless Leadership Summit, June 2004: Convergence of Telecom and Datacom creating a clash	Yankee Group, Совещание лидеров беспроводной связи, июнь 2004: Конвергенция Telecom и Datacom, создающая конфликт
Applications and Services Messaging, mCommerce, Gaming, Content etc	Приложения и услуги Обмен сообщениями, м-коммерция, игры, содержания, и т.д.
Middleware e.g. Web services	Промежуточное ПО напр., услуги Web
Service Platform	Платформа услуг
Middleware (e.g. OSA/Parlay)	Промежуточное ПО (напр., OSA/Parlay)
Service Enabling System	Система обеспечения услуги
Network	Сеть

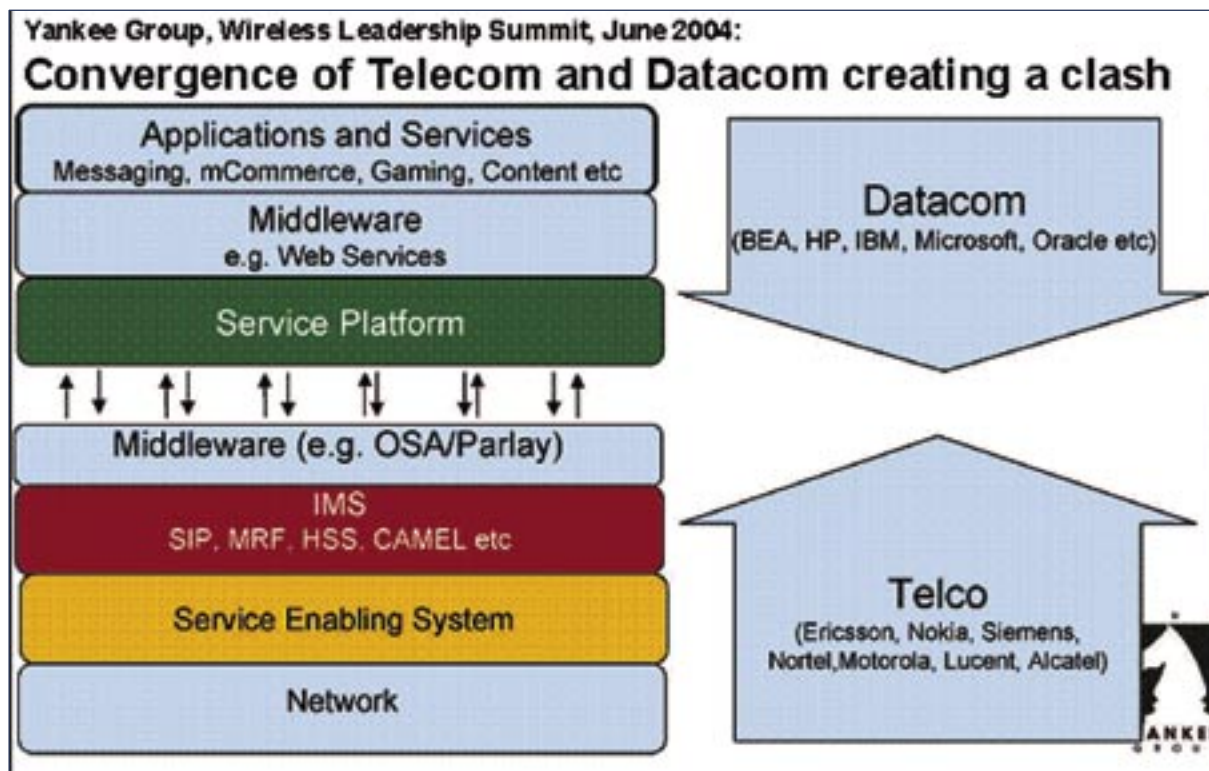


Рис. 12. Конвергенция Telecom и Datacom

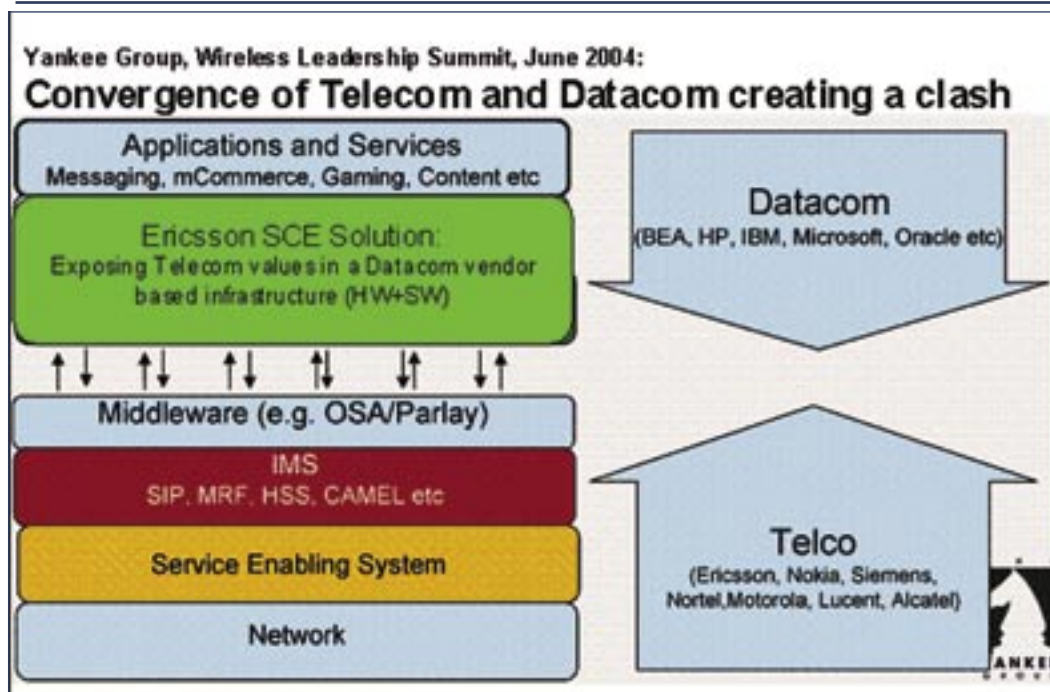


Рис. 13.
Позиция
окружения для
развития услуг
в контексте
конвергенции
telecom и
datacom

Yankee Group, Wireless Leadership Summit, June 2004: Konvergence of Telecom and Datacom creating a clash	Yankee Group, Совещание лидеров беспроводной связи, июнь 2004: Конвергенция Telecom и Datacom создающая конфликт
Applications and Services Messaging, mCommerce, Gaming, Content etc	Приложения и услуги Обмен сообщениями, м-коммерция, игры, содержания, и т.д.
Ericsson SCE Solution: Exposing Telecom values in a Datacom vendor based infrastructure (HW+SW)	Решение SCE Эрикссона: Воздействие Telecom значений на Datacom инфраструктуру, базиро- ванную на поставщиках (HW+SW)
Middleware (e.g. OSA/Parlay)	Промежуточное ПО (напр., OSA/Parlay)
Service Enabling System	Система обеспечения услуги
Network	Сеть

услуги, транслирующие системы для (синтаксической) адаптации и (семантического) посредничества в направлении различных информационных и коммуникационных стандартов, а также шлюзов, содержащих управление надежностью доступа, и т.д.;

- поддержка базовой логики (Core Logic Support), наряду с поддержкой для остальных слоев: система гармонизации, определяющая последовательность деловых процессов (orchestration/work flow engine), система правил (rule engine), поддержка для транзакций, синхронная и асинхронная коммуникация сообщениями, и т.д.;
- поддержка для соединения с ресурсами (Resource Connector Support): транслирующие системы для синтаксической адаптации и семантического посредничества, и т.д.

Прикладной сервер, удовлетворяющий J2EE спецификации прикладной платформы, является примером платформы, которая обеспечивает высокую степень поддержки самому прикладному уровню. На Рис.16. дан обзор прикладных серверов, доступных на J2EE платформе.

Особое значение имеет поддержка для XML web услуг, специфицированная в запросах JSR-109 Web Services, JSR-101 JAX-RPC, а также J2EE архитектура соединителя (connector architecture), специфицированная в запросе JSR 112 JCA.

4.2.3. Автоматизация деловых процессов

Большинство компаний использует прикладную поддержку для автоматизации все большего числа своих деловых процессов (work flows). Такие процессы состоят из подпроцессов/активностей в роли компонентов. Со временем, параллельно с изменениями деятельности открываются новые возможности, а некоторые старые

закрываются, что ведет к насущной потребности приспособления новым запросам деятельности быстрым и эффективным способом.

Для постижения такой функциональности, важно опознать и изолировать стабильные деловые процессы от тех процессов, которые в большей степени подвержены изменениям. Стабильные деловые процессы должны быть реализованы как компоненты, которые предоставляют услуги по четко определенным договорам. Части, подверженные изменениям, должны быть в состоянии меняться наряду с быстрой и простой комбинацией компонентов в новые процессы.

Языки, инструменты и окружение для выполнения и гармонизации деловых процессов уже находятся на рынке. Язык для выполнения деловых процессов для web услуг (BPEL4WS - Business Process Execution Language for Web Services) возник в результате совместной инициативы компаний BEA, IBM и Microsoft, и в настоящее время находится в процессе стандартизации OASIS WSBPEL TC, группы занимающейся разработкой языков разметки. Другой совместной инициативой компаний BEA и IBM является язык для выполнения деловых процессов для технологии Java (BPELJ - Business Process Execution Language for Java), конкретизированный в запросе для спецификации Java номер 207, (JSR - Java Specification Request).

Деловые процессы могут быть описаны двумя способами. Исполнительные деловые процессы моделируют действительное поведение участников деловой интеракции. Деловые протоколы, в свою очередь, используют описания процессов, которые специфицируют поведение взаимно видимых сообщений для каждой стороны, включенной в протокол, без открытия внутреннего поведения.

Описания процессов для деловых протоколов

называются абстрактными процессами. Язык BPEL4WS служит для моделирования поведения исполнительных и абстрактных процессов.

Язык BPEL4WS служит для формальной спецификации деловых процессов и протоколов деловых интеракций. Таким образом, язык BPEL4WS расширяет модель интеракций web услуг и ясно определяет модель совместимости, которая использует расширения автоматизации деловых процессов внутри самой корпорации, но также и внутри области коммуникации бизнес для бизнеса (B2B).

4.2.4. Архитектура, базирующаяся на компонентах

Для быстрой реакции на новые запросы, задаваемые к приложениям программной поддержки, не достаточно лишь сосредоточиться на гармонизацию деловых процессов, нужно также распознать, ясно определить и разработать компоненты, обеспечивающие услуги для деловых процессов (Component Based Architecture – архитектура, базирующаяся на компонентах).

Чтобы добиться высокой эффективности при развитии и реализации компонентов, компоненты должны быть разработаны в соответствии с хорошо определенной моделью компонентов. Модель компонентов специфицирует правила, которые должны соблюдать сами компоненты.

J2EE представляет такую модель компонентов и в этой функции является опорой для развития услуг.

4.3.5. Договоры и архитектура, ориентированная на услуги

Частью модели компонентов является договорная модель, которая специфицирует правила, которые должны соблюдаться для

успешного экспонирования и использования услуги, предоставляемой самим компонентом (Contracts and Service Oriented Architecture – архитектура, ориентированная на договоры и услуги). Такая договорная модель может, в большей или меньшей степени, зависеть от окружения/платформы и поддерживать свободные связи (loose coupling) между различными сторонами, представленными договором на различных уровнях.

Архитектура, ориентированная на услуги (SOA - Service Oriented Architecture), относится к категории договорных моделей, которые не зависят от окружения/платформы самого компонента, а значит, не зависят ни от реализации самого компонента.

Архитектура SOA также делает акцент на прозрачности, поэтому SOA нужно было бы применять во всех тех случаях, где есть потребность в свободных связях между участниками договора, что особенно благоприятно для соединений между слоями в распределенной прикладной поддержке.

Целью окружения для создания услуг является улучшение совместимости при взаимодействии деловых субъектов, что накладывает потребность в более эффективных и надежных совместимых технологиях, базирующихся на принципе SOA.

Одной из таких технологий являются web услуги (WS – Web Services), которые специфицировал Консорциум W3C (World Wide Web Consortium). Процесс использования web услуг между заинтересованными сторонами представлен на Рис.17.

Технология Web услуги приобрела широкую поддержку

Client domain/tier of the system domain in focus	Клиентский слой целевого домена системы
System domain in focus	Целевой домен системы
Presentation Facade tier	Представительский слой фасада
Core Logic tier	Слой базовой логики
Resource Connector tier	Слой соединителя ресурсов
Resource domain/tier of the system domain in focus	Слой ресурсов целевого домена системы

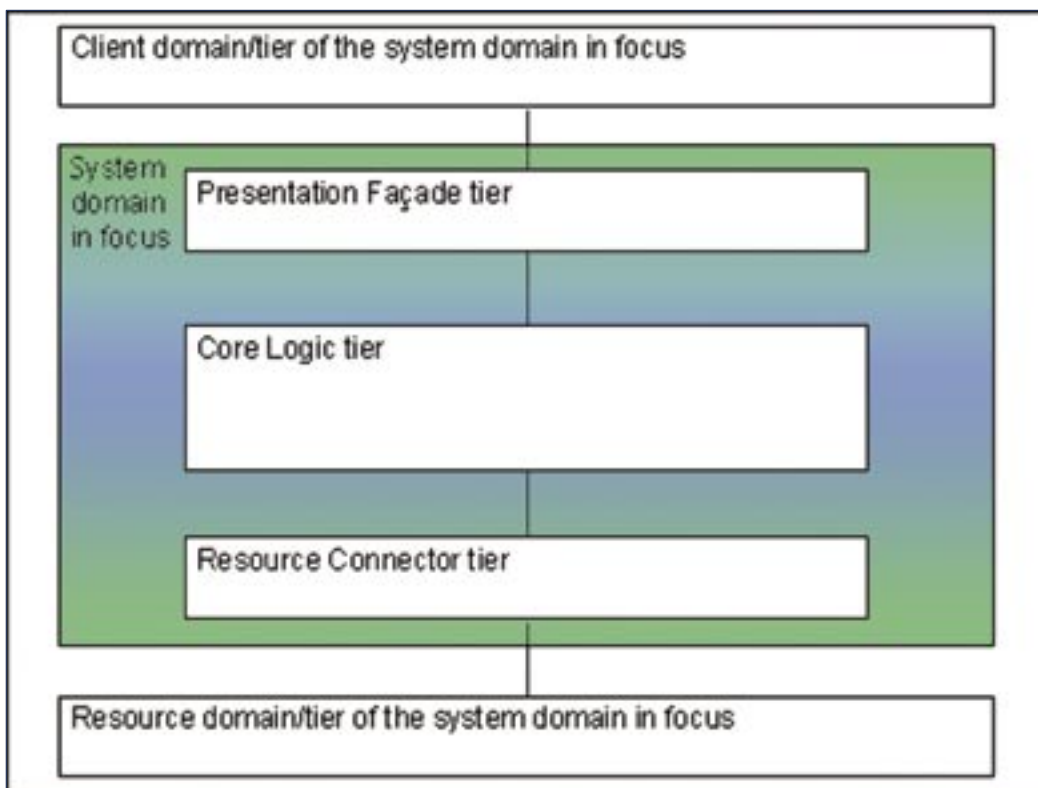


Рис. 14. Обзор общего слоя домена прикладных систем

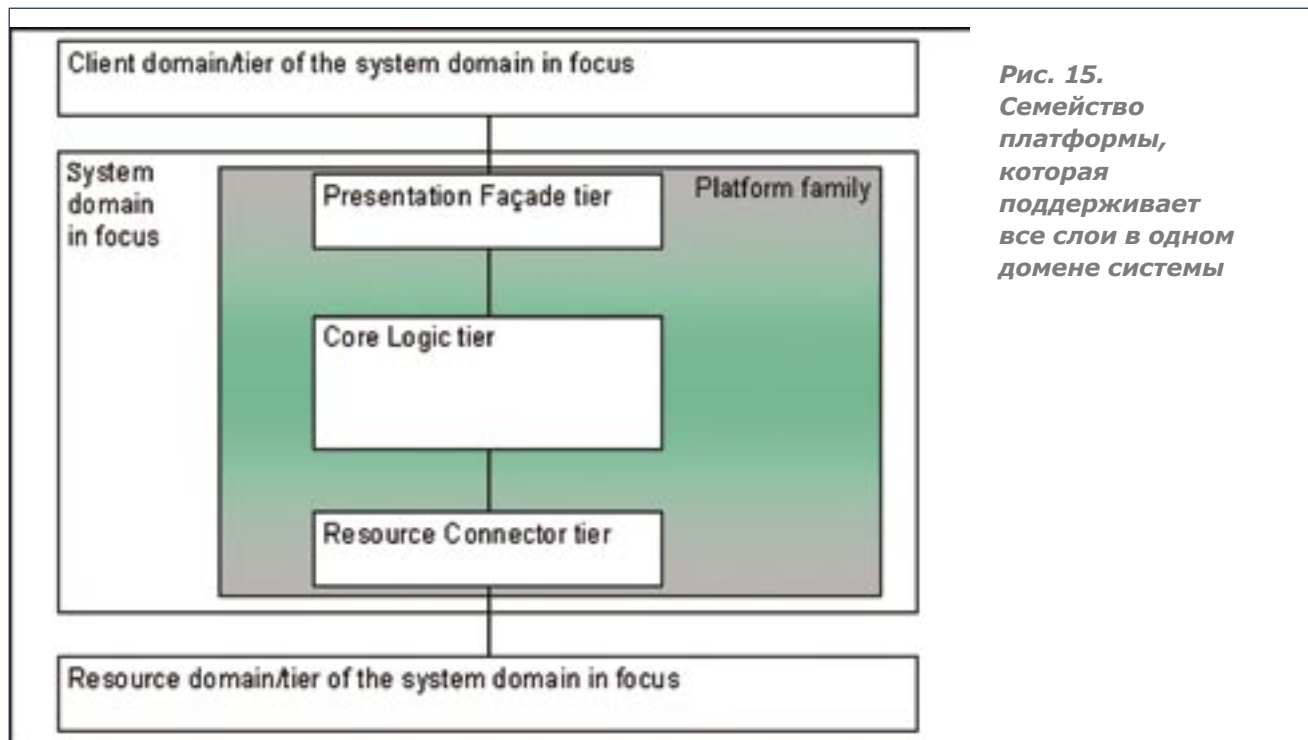


Рис. 15.
Семейство платформ, которая поддерживает все слои в одном домене системы

Client domain/tier of the system domain in focus	Клиентский слой целевого домена системы
System domain in focus	Целевой домен системы
Presentation Façade tier	Представительский слой фасада
Platform family	Семейство платформ
Core Logic tier	Слой базовой логики
Resource Connector tier	Слой соединителя ресурсов
Resource domain/tier of the system domain in focus	Слой ресурсов целевого домена системы

как важная реализация архитектуры, ориентированной на услуги. Причиной этому являются возможности web услуг, позволяющие интеграцию крайне неоднородных приложений поверх сети Интернет, применяя распределенную вычислительную модель. Спецификация web услуг в целом независима от программных языков, операционных систем, а также самих платформ, и стимулирует свободную связь между пользователем и поставщиком услуг. Технология базируется на открытых технологиях, перечисленных ниже:

- eXtensible Markup Language (XML) – расширяемый язык разметки;
- Simple Object Access Protocol (SOAP) – простой протокол доступа к объектам;
- Universal Description, Discovery and Integration (UDDI) – универсальное описание, поиск и взаимодействие, стандарт UDDI;
- Web Services Description Language (WSDL) – язык описания Web услуг.

Использование открытых стандартов обеспечивает широкие возможности совместимости решений различных поставщиков. Этот принцип позволяет компаниям внедрение web услуг без малейшего знания о пользователе услуг, и наоборот. В конечном счете, это обеспечивает простую и быструю интеграцию, поддерживающую несложное и динамическое установление новых деловых партнерств.

Кроме основной спецификации web услуг, развивается и группа передовых WS-* спецификаций, таких как, например, адресация web услуг (WS-addressing), надежность web услуг (WS-Security), безотказный обмен сообщениями (WS-ReliableMessaging), которые

совместно разработали компании Microsoft, BEA и IBM, а затем передали их организациям по стандартизации (прежде всего, W3C и WS-I), для обеспечения их совместимости.

Группа протоколов для web услуг (Web Service Protocols) разработана в форме семейства блоков, базирующихся на модульной архитектуре SOAP. Профили спецификаций выполнены таким образом, что формулируют ограничения использования и предлагают самые лучшие искусства для использования спецификаций в различных комбинациях.

4.2.6. Совместная информационная модель

Для решений, в которых взаимодействует много прикладных систем/компонентов, самым простым способом для обеспечения совместимости является создание совместной информационной модели (Shared Information Model). Информационная модель, или части этой модели, могут затем использоваться во многих договорах услуг. Таким образом, создается семейство договоров услуг, определенных согласно одной информационной модели. Если не существует совместной информационной модели, навязывается потребность комплексных транзакций/адаптаций и посредничества информации. Кроме того, если не существует совместной информационной модели, появляется риск от невозможности поддержки и автоматизации деловых процессов.

Совместная информационная модель не значит, что все сведения доступны и известны всем системам/компонентам, а это также не подразумевает сохранение всех сведений в общей базе.

Однако типы атрибутов в специфичном информационном домене обычно связаны с общими типами объектов, которые являются совместными для большего числа информационных доменов. Например, такие общие объекты, как “пользователь”, “услуга”, и т.п., обладают различными группами атрибутов, связанных с доменами оплаты, активирования и обеспечения услуги, однако существенным является то, что каждый

из информационных доменов использует совместную информационную модель общего типа объекта.

При отсутствии такой модели тяжело отображать (mapping) информацию или акции между различными доменами. Предпочитается внимательное определение совместной группы информации, используемой в деятельности/системах, и выделение владельцев, которые контролируют обновление информации и договоров услуг для манипулирования информацией. Кроме того, решающей является возможность простого и безболезненного расширения совместной информационной модели.

4.2.7. Архитектура, базирующаяся на правилах управления

При поставке информации и коммуникационных услуг, задаются определенные правила управления (Policy Enabled Architecture), сформулированные для выполнения различных деловых, системных или других целей.

Если правила управления для приложений, поставляющих услуги, определены заранее и никогда не будут меняться, тогда можем создать неизменяемую (hard-coded) прикладную поддержку.

Однако это не так, и навязывается потребность установки и применения правил управления в течение самого выполнения. В архитектуре, базирующейся на правилах управления, образец "разделения интересов" применен еще раз, а в цепи поставки услуги определены две роли: точка приведения в исполнение правил управления (PEP - Policy Enforcement Point) и точка принятия решения об управлении (PDP - Policy decision Point).

Роль точки принятия решения об управлении выделена центральной прикладной системе, где могут определяться правила управления, и где логика для принятия решений выполняется по требованию прикладных систем, которым присвоена роль точки

приведения в исполнение правил управления. Роль точки приведения правил управления в исполнение выделена прикладной системе, которая служит для проверки, и выполнения определенных правил управления в специфических точках цепи поставки услуг.

Одной из категорий управления является политика надежности. В процессе автоматизации деловых процессов и открытия в направлении B2B совместимости с различными уровнями доверия, наличие постоянного механизма надежности имеет немаловажное значение. Постоянность механизма отражается в возможности исполнения потребностей надежности (и конфиденциальности) всех заинтересованных сторон, а не в неизменяемости установленных отношений, как это, может быть, предполагается на первый взгляд. Наоборот, должна существовать возможность быстрой и эффективной модели управления политикой надежности. Управление политикой надежности является свойством точки PDP. Точка PEP для обеспечения надежности поставлена во встречных точках цепи поставки услуги, обычно в межсетевых шлюзах.

Другая категория политики относится на договоры об уровне услуги и об оплате.

4.2.8. Окружение для развития услуг и технологии телекоммуникаций

В предыдущих главах статьи объяснены отношения окружения для создания услуг в контексте межпромышленных прикладных доступов и технологий, которые все более применимы в домене телекоммуникаций. Данная глава объясняет аспекты, специфичные для телекоммуникаций в контексте самого места обитания окружения для создания услуг (SCE and telecom technologies).

В данный момент существует большое число технологий

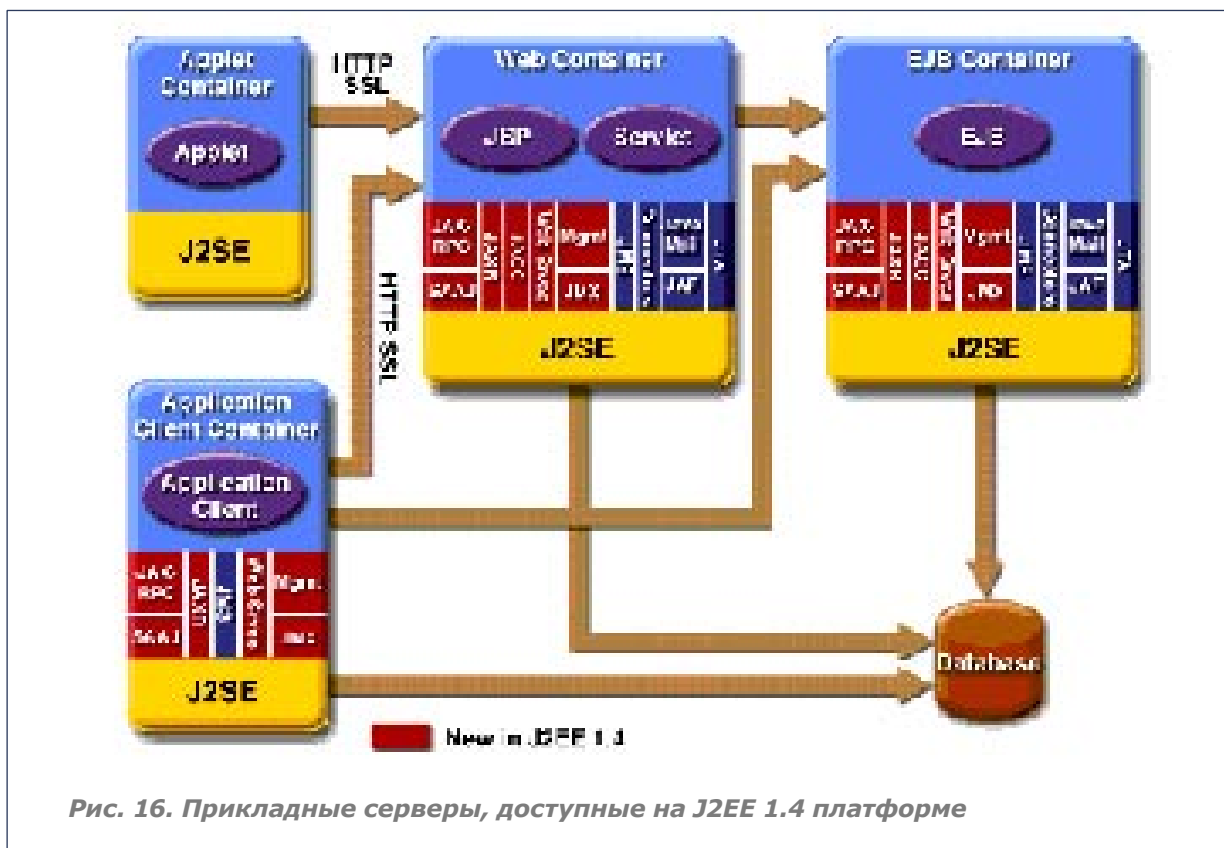


Рис. 16. Прикладные серверы, доступные на J2EE 1.4 платформе

для сетей доступа и опорных сетей. Возникновение новых и исчезновение существующих технологий естественный процесс, направленный на создание простых технологий (преобладающая тенденция) и обеспечение конвергенции различных сетевых технологий. Это особенно важно для технологий опорной сети, которые, кроме всего прочего, играют роль соединителя различных технологий доступа.

Мы можем быть уверены, что в переходном периоде нас ожидает смесь различных генераций телеком

технологий и потребность обеспечения совместимости между различными сетевыми технологиями (сети с коммутацией пакетов PS и с коммутацией каналов CS, сотовые и стационарные сети, сотовые сети второй и третьей генераций - 2G, 2,5G, 3G, и т.п.).

4.3.9. Тенденции телеком технологии из перспективы приложений

Для простой коммуникации между отдельными пользователями (P2P – Person to Person) конвергенция и/или совместимость решены в опорной сети, однако окружение для создания услуг, прежде всего, сосредоточивает внимание на коммуникацию пользователя в направлении содержаний/приложений (P2C&A – Person to Content & Application), и на коммуникацию приложений/содержаний в направлении пользователя (C&A2P – Content & Application to Person).

Освобождение разработчиков приложений от обязанности обладания специфическими знаниями о телекоммуникациях и технологиях, а также избегание повторной разработки уже существующих приложений для различных телеком технологий, потребовало введения шлюзов и посредников, базирующихся на IP технологии и технологически нейтрально представляющих способности сети. Одной из таких инициатив является открытый доступ услуге – интерфейс OSA (Open Service Access), и окружение, определенное группой Parlay. Организация “Проект сотрудничества в создании сетей третьей генерации”, 3GPP, также участвует в разработке OSA спецификаций,

Requester Entity	Запрашивающая сторона
Provider entity	Поставщик
Discovery process(1) in more detail	Процесс опознавания (1), детальнее
Criteria	Критерии
Discovery Service	Услуга опознавания
1. Parties "become known" to each other	1. Стороны "познакомились" друг с другом
Requester Entity	Запрашивающая сторона
Provider entity	Поставщик
2. Agree on semantics & WSD	2. Договор о семантике и WSD
Requester Human	Запрашивающая сторона - человек
3. Input Semantics & WSD	3. Введение семантики и WSD
Requester Agent	Запрашивающая сторона - агент
4. Interact	4. Взаимодействие
Provider Human	Поставщик - человек
3. Input Semantics & WSD	3. Введение семантики и WSD
Provider Agent	Поставщик - агент

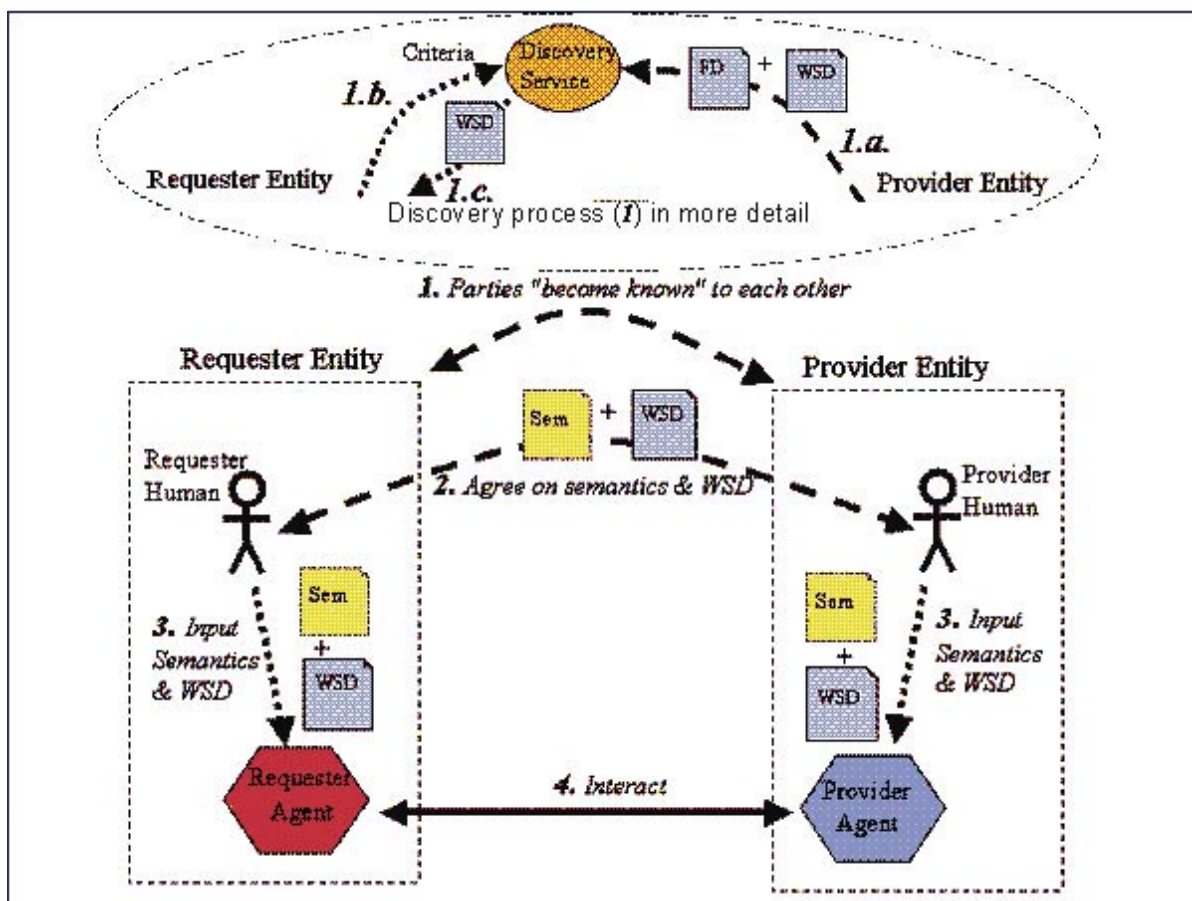
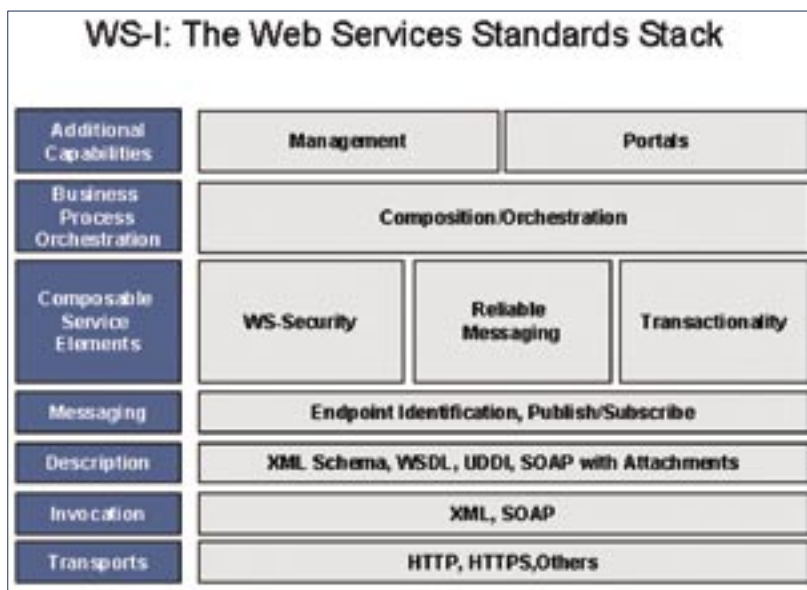


Рис. 17. Процесс использования web услуг



которые и сама усвоила. На Рис.19. представлены упрощенные иллюстрации из 3GPP TS 23.198 и спецификации OSA интерфейсов и окружения.

Инициальные интерфейсы, которые определила группа Parlay, базированы на технологии CORBA. Постепенно группа Parlay приняла web услуги как технологию и объявила ряд спецификаций, основывающихся на web услугах, под названием Parlay X. Целью интерфейсов Parlay X является облегчение развития приложений (Parlay X также признала и организация 3GPP). Parlay X также предпочитаемая технология для открытия сетевых возможностей третьим сторонам, в данном случае посредством шлюзов обеспечения надежности.

Другие инициативы, направленные на приложения и механизмы их создания (enablers), упоминает организация Open Mobile Alliance (OMA). OMA архитектура основывается на следующих принципах:

- Принцип #1 (Principle #1) – создание спецификаций для механизмов создания приложений (enabler), независимых от существующих сетей, операционных систем и программных языков;
- Принцип #2 (Principle #2) –

Рис.18. Стандарты для web услуг

WS-I: The Web Services Standards Stack	WS-I: Стек стандартов Web услуг
Additional Capabilities	Дополнительные возможности
Business Process Orchestration	Гармонизация деловых процессов
Composable Service Elements	Компонуемые элементы услуги
Messaging	Обмен сообщениями
Description	Описание
Invocation	Инициирование
Transports	Транспорт
Management	Управление
Portals	Порталы
Composition/Orchestration	Формирование/Гармонизация
WS-Security	WS-надежность
Reliable Messaging	Безотказный обмен сообщениями
Transactionality	Трансакции
Endpoint Identification, Publish/Subscribe	Опознавание конечной точки, публичной/абонентской
XML Schema, WSDL, UDDI, SOAP with Attachments	XML схема, WSDL, UDDI, SOAP с приложениями
XML, SOAP	XML, SOAP
HTTP, HTTPS, others	HTTP, HTTPS, другие

discovery	опознавание
framework	структура
Open Service Access	Открытый доступ услуге
OSA	OSA
User Location Call control	Управляющий вызов для локализации пользователя
Service capability server(s)	Серверы возможностей услуги
Application server	Сервер приложений
Application	Приложение
OSA API	OSA API
Interface class	Класс интерфейса
E.g. Location server MExE server SAT server	Напр., Сервер локализации MExE сервер SAT сервер

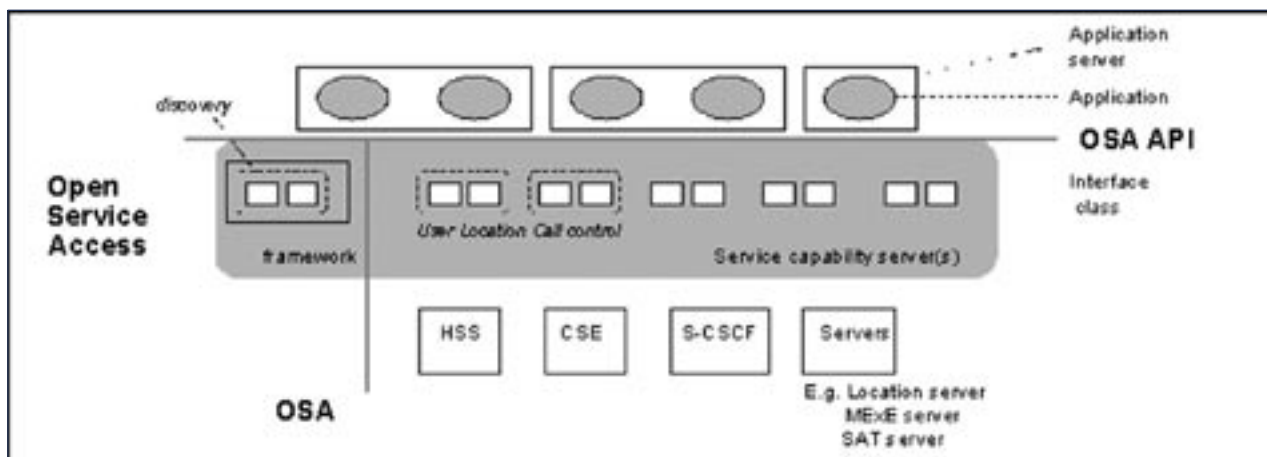
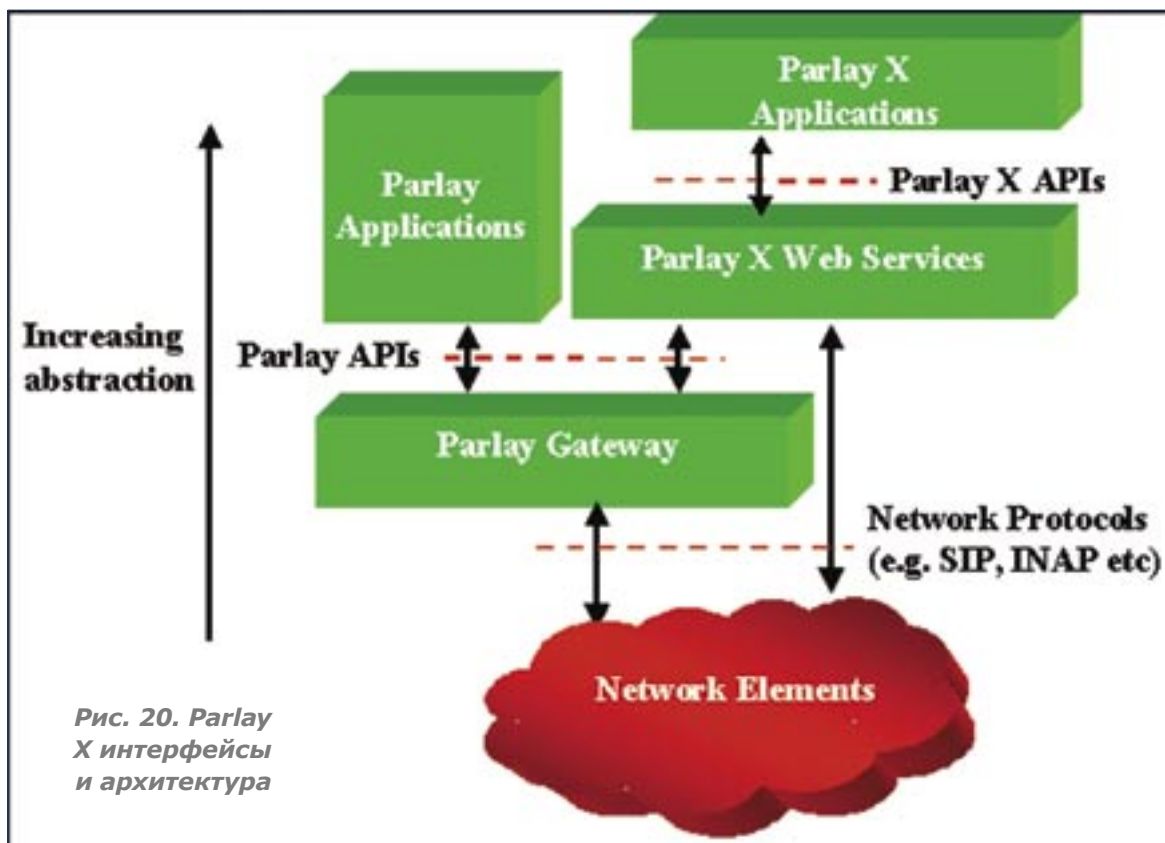


Рис. 19. OSA окружение и интерфейсы согласно 3GPP



Parlay Applications	Parlay приложения
Parlay X Applications	Parlay X приложения
Parlay X APIs	Parlay X APIs
Increasing abstraction	Возрастающая абстракция
Parlay Gateway	Parlay шлюз
Network Protocols (e.g. SIP, INAP etc)	Сетевые протоколы (напр., SIP, INAP и др.)
Network Elements	Элементы сети

• Принцип #5 (Principle #5) – обеспечение надежности, согласованной с моделями надежности в сети Интернет, и конфиденциальности пользователя.

Организация OMA также скицировала OMA окружение услуг (OSE - OMA Service Environment), как представлено на Рис. 21. Такое решение применимо для архитектуры окружения для создания услуг в деловом домене оператора (т.е. домене поставщика каналов), что на Рис. 21. обозначено как домен поставщика услуг (Service Provider Domain).

Кроме того, окружение для развития услуг, прежде всего, предусмотрено для создания предварительных условий сетевым операторам, которые устанавливают партнерские отношения с большим числом партнеров (mass partnering), что навязывает вопрос о конфиденциальности пользовательских данных, определенных в стандартах Проекта свободного объединения, LAP (Liberty Alliance Project). Обзор LAP стандартов дан на Рис. 22.

4.3.11. Тенденции технологии из перспективы механизма для создания приложений (enabler)

Parlay/Osa, OMA и аналогичные инициативы, очевидно, направлены на облегчение развития приложений. Однако задается вопрос, какие тенденции преобладают в опорной части сети, и каким образом они влияют на приложения и доступ к ним. Ответ на этот вопрос

находим в том факте, что базовые части инфраструктуры телекоммуникаций все больше, или даже в целости, полагаются на IP технологии. Главная инициатива в опорной сети относится к мультимедийной подсистеме, базирующейся на Интернет протоколе (IMS - IP Multimedia Subsystem), которую определила организация 3GPP (TS 23.228). Подсистема IMS сфокусирована на мультимедийное управление вызовом, основывающееся на сессии и протоколе инициации сессии (SIP - Session Initiation Protocol).

Главной задачей подсистемы IMS является мультимедийное управление сессией между пользователями (P2P), поддерживаемое приложением в оборудовании пользователя (UE - User Equipment). Однако из аспекта окружения для создания услуг, организация 3GPP определяет IMS окружение услуг/приложений для приложений, размещенных в прикладных серверах (AS - Application Servers), (Рис. 23).

Приложения общаются с опорой IMS подсистемы посредством:

- интерфейса управления услугой, ISC (IMS Service Control), базированного на протоколе SIP, в направлении функции управления вызовом/сессией, S-CSCF (Serving Call/Session Control Function);
- интерфейса Si/Sh, базированного на протоколе Diameter, в направлении бытового абонентского сервера (HSS - Home Subscriber Server).

Подсистема IMS организации 3GPP определяет три различные категории IMS окружений услуг/приложений:

- функция IM SSF (IP Multimedia Switching Service Function) – традиционные приложения (Legacy Applications);
- SIP прикладной сервер (SIPAS) – исходные IMS приложения;
- сервер OSA SCS (OSA Service Capability Server) – возможностей услуг, предлагаемых посредством OSA окружения в направлении OSA прикладных серверов.

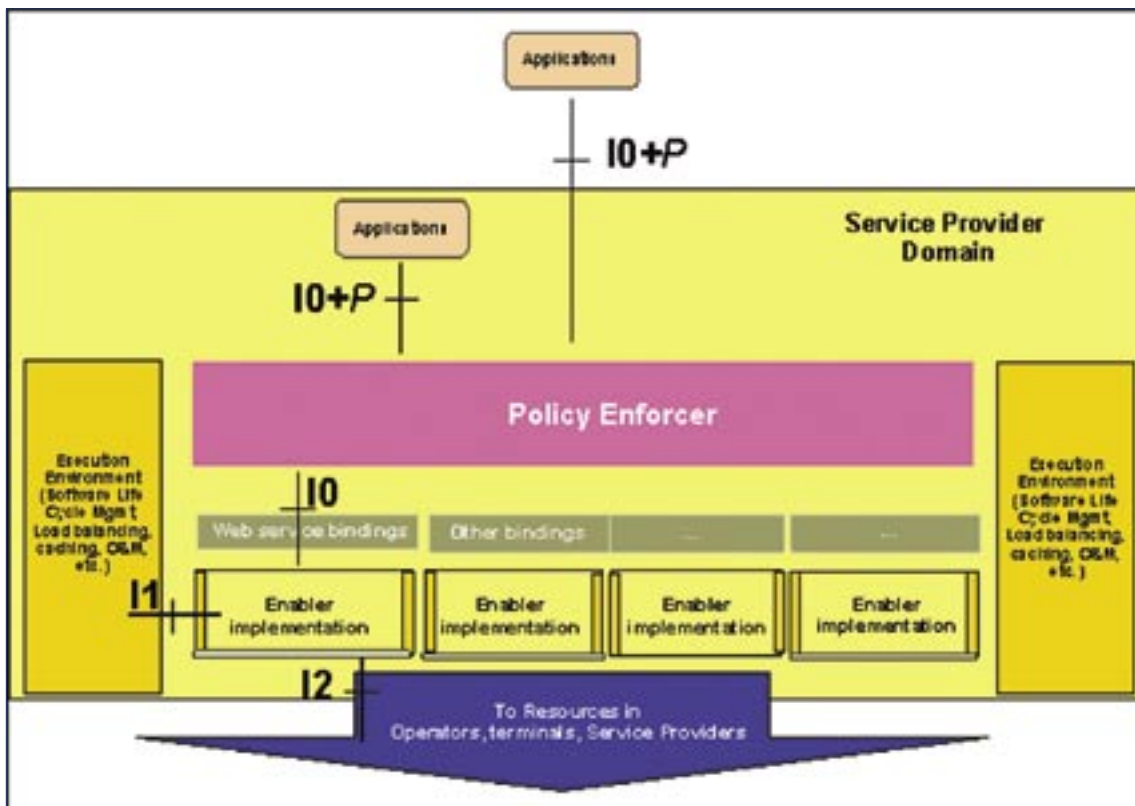


Рис. 21. ОМА окружение услуг (OSE) – концептуальная архитектура и категории интерфейсов

Поддержка приложениям, таким как, присутствие и управление группами пользователей, специфицирована для сохранения в прикладных серверах, базирующихся на протоколе SIP. Все это примеры общих характеристик, которые дополняют мультимедийную сессию посредством протокола SIP и его расширений. В основном, направлены на поддержку приложений на конечном оборудовании пользователя, но и приложения, размещенные в SIP прикладных серверах, также могут использовать эти характеристики.

Поддержка, базирующаяся на протоколе SIP, не единственная характеристика, требуемая от приложений. Для пользовательских приложений существуют также другие интерфейсы и протоколы, такие как, например, интерфейс IMS Ut (базирующийся на XCAP и HTTP) для манипулирования данными в прикладной поддержке,

Applications	Приложения
Service Provider Domain	Домен поставщика услуги
Policy Enforcer	Приведение в исполнение
Web service bindings	Web компоновки услуг
Other bindings	Другие компоновки
Enabler implementation	Enabler, внедрение
To Resources in Operators, terminals, Service Provider	К ресурсам оператора, терминалов, поставщика услуг
Execution Environment (software Life Cycle Mgmt Load balancing, caching, O&M, etc.)	Условия выполнения (жизненный цикл программ, Mgmt балансировка загрузки, кэширование, O&M, и т.д.)

размещенной в SIP прикладных серверах.

Для приложений, размещенных в прикладных серверах, и находящихся в фокусе окружения для создания услуг, направляется потребность в интерфейсах,

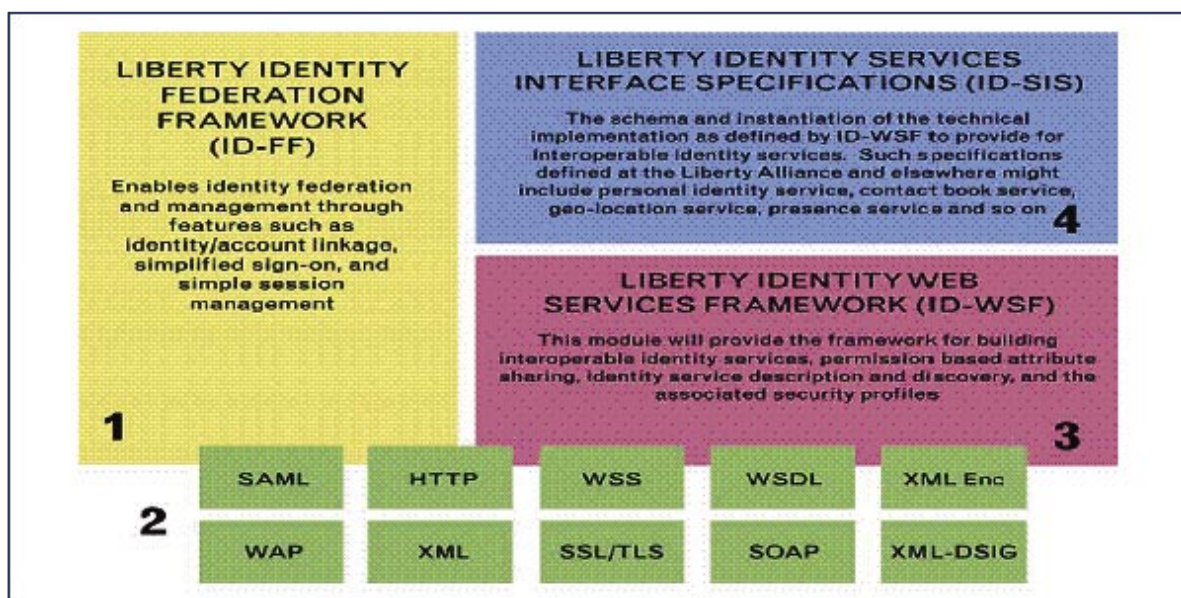


Рис. 22. Обзор стандартов, прописанных организацией IAP

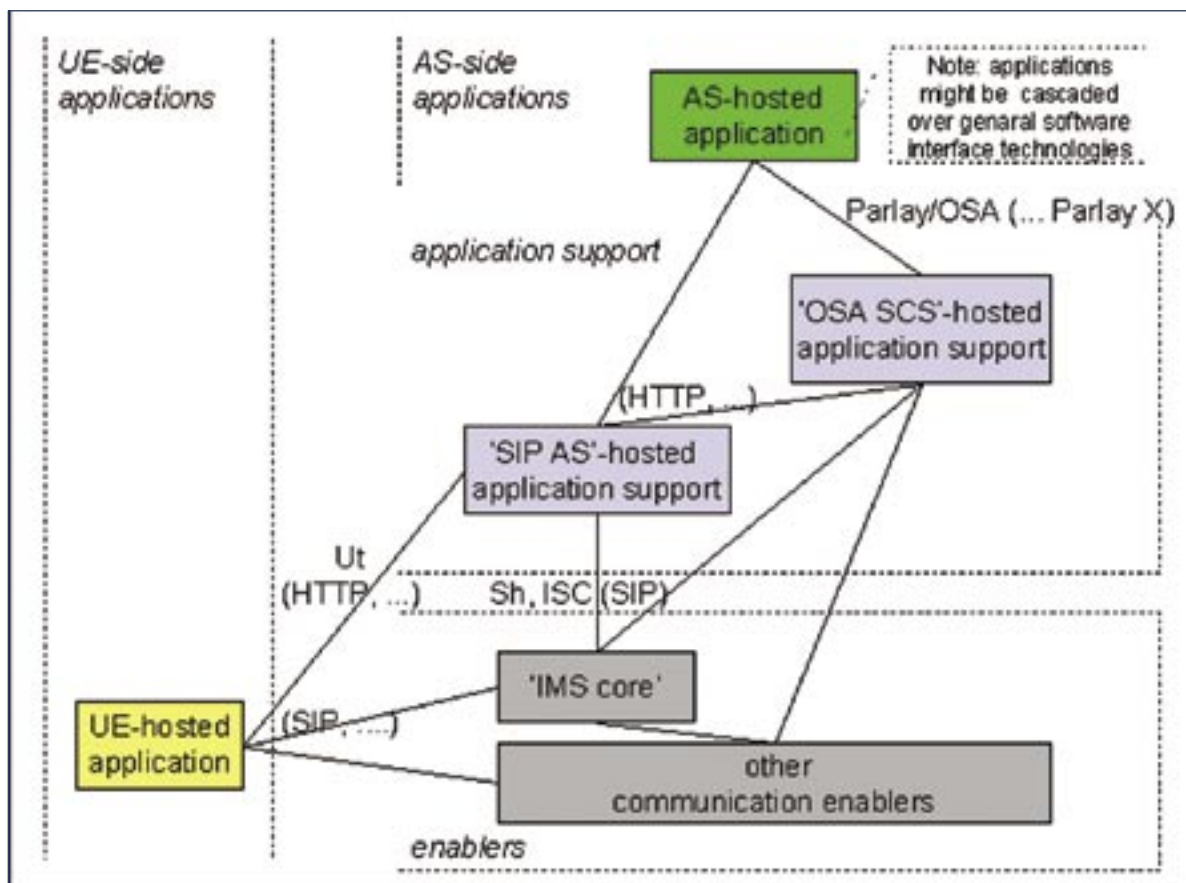


Рис. 23. Приложения, поддержка приложениям и механизмы создания приложений (enablers)

UE-side applications	Приложения стороны UE
UE-hosted application	Приложение, размещенное в UE
AS-side applications	Приложения стороны AS
AS-hosted application	Приложение, размещенное в AS
application support	поддержка приложения
Note: applications might be cascaded over general software interface technologies	Сноска: приложения могут быть каскадными через интерфейс основного ПО
OSA SCS – hosted application support	Поддержка приложения, размещенного в OSA SCS
SIP AS – hosted application support	Поддержка приложения, размещенного в SIP AS
IMS core	Опора IMS
other communication enablers	другие механизмы коммуникации
enablers	enablers

которые превосходят возможности SIP протокола (и его расширений), а также в характеристиках, обеспечиваемых подсистемой IMS. Например, интерфейс Parlay/OSA предоставляет большой диапазон способностей поддержки приложениям, размещенным в прикладных серверах, и открывает новые возможности конвергенции телеком технологий из аспекта приложения, размещенного в прикладном сервере. Кроме того, интерфейс Parlay/OSA обеспечивает возможность предоставления телеком способностей третьим сторонам посредством web услуг и шлюзов обеспечения надежности, а также вводит возможности контроля и приведения в исполнение договора об уровне услуги. Предпочитаемой технологией окружения для развития услуг является технология J2EE. Здесь напрашивается потребность рассмотрения и включения SIP прикладного сервера в платформу J2EE. Поддержка технологий,

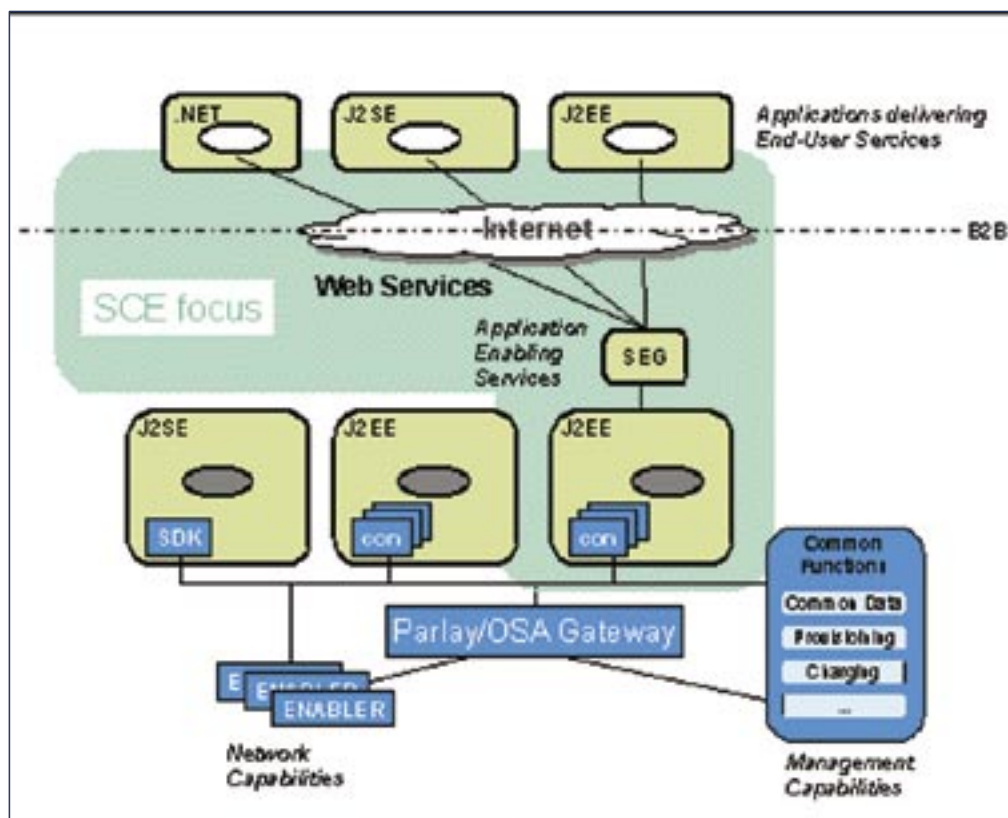
базирующихся на протоколе SIP, реализована как сервлет. Сервлеты реализованы в SIP контейнере сервлетов, который поддерживает и SIP, и HTTP коммуникацию, как это представлено на Рис. 24.

4.3.12. Окружение для развития услуг в контексте телеком технологий и прикладной поддержки

Если комбинируем подход экспонирования сетевых способностей с межпромышленными прикладными окружениями для развития приложений и B2B совместимостью, получаем ситуацию, представленную на Рис. 25. (cop= соединитель к механизмам создания приложений /ресурсам; SEG = шлюз обеспечения надежности).

Этот контекст является основой окружения для создания услуг, с помощью которого обеспечены услуги приложениям. Соответствующие административные услуги, предоставляемые оператором (который действует как поставщик каналов), даны в распоряжение поставщиков услуг с дополнительной стоимостью (VASP - Value Added Services Providers) посредством web услуг и шлюзов обеспечения надежности. Это обеспечивает эффективное развитие приложений, при чем действительная комплексность технологий телекоммуникаций остается невидимой. Если сосредоточим внимание на существующее окружение для создания услуг и на его перспективу, на решение, которое успешно внедряет предыдущие и будущие технологии в совместное окружение, получим

Рис. 25. Типичное решение окружения для развития услуг в контексте межпромышленного прикладного окружения и телеком окружения



Applications delivering End-User Services	Приложения поставляющие услуги конечным пользователям
Internet	Интернет
Web services	Web услуги
SCE focus	SCE фокус
Application Enabling services	Услуги позволения приложения
Common functions	Общие функции
Common Data	Общие данные
Provisioning	Инициализация
Charging	Оплата
Management Capabilities	Возможности управления
Network capabilities	Возможности сети

- Разработчик приложения может разрабатывать приложения без изучения способа и модели оплаты (в этом случае прикладное окружение поддерживает опознавание поставленных услуг); и/или
- Разработчик приложения может использовать услуги для обеспечения приложений для оплаты, без знания правил оплаты услуг, с тем, что о выполнении услуги может быть информировано приложение для оплаты; и/или
- Разработчик приложения может использовать услуги для оплаты без знания правил оплаты, но он должен принимать во внимание занятость средств/ресурсов.
- Разработчик приложения и собиратель содержаний и приложений используют услуги оплаты за услуги гибким способом.

3. Расходы деятельности минимизированы для каждой услуги

Параллельно с ростом числа предлагаемых услуг цена жизненного цикла услуги увеличивается, а значит, увеличиваются расходы ежедневной деятельности.

- Оператор может эффективно управлять предложением своих услуг разработчикам приложений, поставщикам приложений и собирателям содержаний и приложений, что влияет на снижение расходов.

- Оператор (поставщик каналов) может предоставить административные услуги за использование услуг посредством порталов, используемых остальными заинтересованными в цепи предложения.
- Оператор (поставщик каналов) может предложить маркетинговый канал посредством самообслуживаемых порталов к конечным пользователям.

4. Отличная поддержка взаимодействия с деловыми партнерами

Все больше партнеров подчеркивают потребность в лучших B2B интерфейсах, базированных на стандартных технологиях, таких как web услуги.

- Оператор (поставщик каналов) может предоставлять услуги большому числу разработчиков приложений, поставщиков приложений и собирателей содержаний и приложений посредством стандартных промышленных интерфейсов, таких как web услуги.
- Разработчики и поставщики приложений, а также собиратели содержаний и приложений могут использовать услуги, предлагаемые операторами (поставщиками каналов), посредством интерфейса B2B, базированных на промышленных стандартах, таких как web услуги.

5. Поддержка стратегии маркетинга, ориентированной на оконечного пользователя

Способы маркетинга должны быть независимыми от доступа и пользовательского оборудования.

- Оператор (поставщик каналов) может предоставлять услуги приложениям, которые или скрывают тип сети доступа и тип оборудования пользователя, или, если этот поступок не подходит данной ситуации, выставляют тип сети доступа и пользовательского оборудования. В первом случае речь идет о включении в приложение адаптации для определенного типа сети доступа и оборудования. Во втором случае речь идет о разделении атрибутов и способностей пользователей ради достижения цели: поставки услуги, независимо от типа сети и оборудования.

• Разработчик приложений, поставщик приложений и сборщик содержаний и приложений может развивать/предоставлять услуги окончному пользователю, независимо от типа сети доступа или оборудования, или с помощью адаптации оператора (как поставщика каналов), или с использованием услуг, которые обеспечивают возможность такой адаптации.

6. Развитие услуг независимо от поставщиков сетевого оборудования

Различия в сетевом оборудовании различных производителей остаются скрытыми для разработчиков приложений.

• Оператор (поставщик каналов) может предоставить услуги для приложений, скрывая сетевое оборудование и их отличия.

• Разработчики приложений могут разрабатывать приложения без знания о специфике сетевого оборудования, требуемого для целостной коммуникации.

• Поставщик приложений может выполнить приложения без знания о специфике сетевых элементов, требующихся для целостной коммуникации.

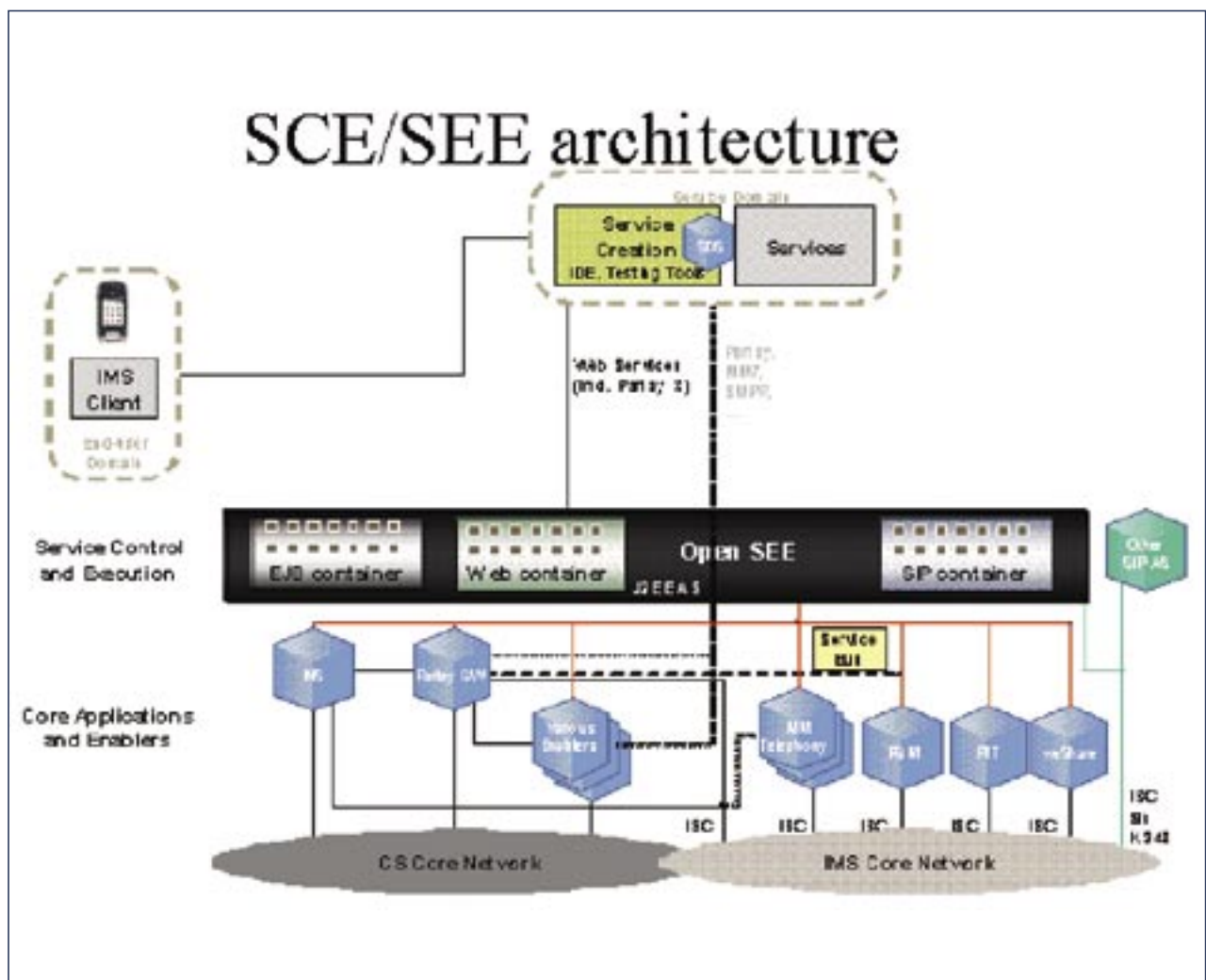


Рис. 26. Конвергентное окружение для создания услуг

6. Список сокращений

AD	-	Application Developer Разработчик приложений	IM SSF	-	IP Multimedia Service Switching Function Функция IP коммутации мультимедийной услуги
AP	-	Application Provider Поставщик приложений	ISC	-	IMS Service Control IMS управление услугой
AS	-	Application Server Сервер приложений	ISV	-	Independent Software Vendor Независимый поставщик программного обеспечения
B2B	-	Business to Business Интерфейс B2B, бизнес для бизнеса	JCA	-	Java Connector Architecture Java архитектура соединителя
BPEL	-	Business Process Execution Language Язык выполнения делового процесса	JSR	-	Java Specification Request Java запрос спецификаций
BPELJ	-	Business Process Execution Language for Java Язык выполнения делового процесса для технологии Java	J2EE	-	Java 2 Enterprise Edition Технология разработки корпоративных приложений на языке Java, вариант 2
BPEL4WS	-	Business Process Execution Language For Web Services Язык выполнения делового процесса для Web услуг	J2SE	-	Java 2 Standard Edition Java 2 стандартное ПО
CAA	-	Content and Application Aggregator Собиратель/агрегатор содержаний и приложений	LAP	-	Liberty Alliance Project Проект свободного объединения
C&A2P	-	Content And Application To Person Содержание и приложение в направлении пользователя	MMS	-	Multimedia Messaging System Система мультимедийных сообщений
CORBA	-	Common Object Request Broker Architecture Общая архитектура посредника запросов к объектам	NP	-	Network Provider Оператор сети
CP	-	Content Provider Поставщик содержаний	OASIS	-	Organization for the Advancement of Structured Information Standards Организация по продвижению стандартов для структурированной информации
EIS	-	Enterprise Information System Информационная система предприятия	OMA	-	Open Mobile Alliance Открытое объединение поставщиков мобильного оборудования
EJB	-	Enterprise Java Beans Спецификация EJB	OSA	-	Open Service Access Открытый доступ услуге
eTOM	-	Enhanced Telecom Operations Map Улучшенное отображение телеком операций	OSA	-	Open Services Architecture Архитектура открытых услуг
EU	-	End User Оконечный пользователь	PDP	-	Policy Decision Point Точка принятия решения об управлении
3GPP	-	3rd Generation Partnership Project Проект сотрудничества в создании сетей третьей генерации	PEP	-	Policy Enforcement Point Точка приведения правил управления в исполнение
HSS	-	Home Subscriber Server Бытовой абонентский сервер	P2P	-	Person to Person Пользователь-Пользователь
HTTP	-	Hyper Text Transfer Protocol Протокол передачи гипертекста	P2C&A	-	Person To Content And Application Пользователь в направлении содержания и приложения
IMS	-	IP Multimedia Subsystem Мультимедийная подсистема, базирующаяся на IP	SCE	-	Service Creation Environment Окружение для создания услуг
			SCS	-	Service Capability Server Функция управления сессией
			S-CSCF	-	Serving Call/Session Control Function Функция управления сессией/обслуживанием вызова
			SDP	-	Service Delivery Platform Платформа поставки услуги

SIP	- Session Initiation Protocol Протокол инициации сессии
SLA	- Service Level Agreements Договоры об уровне услуги
SMS	- Short Message Service Служба кратких сообщений
SNF	- Service Network Framework Структура сети услуг
SOA	- Service Oriented Architecture Архитектура, ориентированная на услуги
SOAP	- Simple Object Access Protocol Простой протокол доступа объекту
TINA-C	- Telecommunications Information Networking Architecture Consortium Консорциум архитектуры объединения в сеть телекоммуникационной информации
TTM	- Time To Market Время от идеи до рынка
UDDI	- Universal Description, Discovery and Integration Универсальное описание, поиск и взаимодействие, стандарт
VAS	- Value Added Service Услуга с дополнительной стоимостью
VASP	- Value Added Services Provider Поставщик услуг с дополнительной стоимостью
W3C	- World Wide Web Consortium Консорциум World Wide Web
WSDL	- Web Services Description Language Язык описания Web услуг
WS-I	- Web Services Interoperability
XCAP	- XML Configuration Access Protocol XML Протокол конфигурации доступа
XML	- Extensible Markup Language Расширяемый язык разметки

И. Барач

e-mail: ivan.barac@ericsson.com

Ericsson Nikola Tesla d.d.

Крапинска 45

р.р. 93

HR-10002 Zagreb

Хорватия

Редакция приняла рукопись 8 ноября 2005.

Перевод: Надежда Племенич.

7. Литература

- [1] Литература компании эрикссон внутреннего использования
- [2] TM Forum NGOSS eTOM (GB921)
<http://www.tmforum.org/browse.asp?catID=1647>
- [3] TM Forum NGOSS Technology-Neutral Architecture
<http://www.tmforum.org/browse.asp?catID=1685>
- [4] TM Forum NGOSS Shared Information/Data Model
<http://www.tmforum.org/browse.asp?catID=1684>
- [5] J2EE
<http://java.sun.com/j2ee/1.4/docs/index.html>
<http://jcp.org/en/jsr/detail?id=151>
<http://jcp.org/en/jsr/tech?listBy=3&listByType=platform>
- [6] Web Services
- W3C <http://www.w3.org/2002/ws/>
- WS-I <http://www.ws-i.org/>
- [7] SOA / Web Services
- BEA <http://dev2dev.bea.com/technologies/soa/index.jsp>
- IBM <http://www-306.ibm.com/software/solutions/webservices/>
- MSFT <http://msdn.microsoft.com/webservices/>,
<http://msdn.microsoft.com/library/en-us/dnanchor/html/WebServicesAnchor.asp>
- Oracle <http://www.oracle.com/technology/tech/webservices/index.html>
- Sun <http://java.sun.com/webservices/index.jsp>
- OMA http://www.openmobilealliance.org/release_program/owser_v10.html
- ZapThink <http://www.zapthink.com/index.html>
- [8] Parlay/OSA specifications
- <http://www.parlay.org/specs/index.asp>
- <http://www.3gpp.org/specs/specs.htm>
TS 23.127, TS 23.198, TS 29.198-xx, TS 29.199-yy, TS 29.998-zz
- [9] OMA (Open Mobile Alliance)
http://www.openmobilealliance.org/release_program/index.html
(see also http://member.openmobilealliance.org/ftp/Public_documents/ARCH/Permanent_documents/)
OMA Architecture Requirements
http://www.openmobilealliance.org/release_program/docs/RD/OMA-RD_Architecture_V1_0-20031021-A.pdf
OMA Service Environment (OSE)
http://www.openmobilealliance.org/release_program/docs/RD/OMA-Service-Environment-V1_0-20040907-A.pdf
- [10] LAP (Liberty Alliance Project) Specifications
<http://www.projectliberty.org/resources/specifications.php>
- [11] BPEL4WS
- BEA <http://dev2dev.bea.com/techtracks/BPEL4WS.jsp>
- IBM <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>
- MSFT <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbizspec/html/bpel1-1.asp>
- OASIS WSBPEL TC http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- [12] Java/J2EE Runtime and Development Environments - examples
- Apache <http://www.apache.org/>
- BEA WebLogic <http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products>
- IBM WebSphere, IBM Rational Software Architect, etc
<http://www-306.ibm.com/software/sw-bycategory/>,
<http://www-128.ibm.com/developerworks/platform/index.html>
- JBoss <http://www.jboss.com/>
- Oracle Application Server <http://www.oracle.com/appserver/index.html>
& Oracle Development Tools <http://www.oracle.com/tools/index.html>