

Тема 7.

**ЭВОЛЮЦИЯ ТЕХНОЛОГИЙ ЭКСПЛУАТАЦИОННОГО
УПРАВЛЕНИЯ СЕТЯМИ И УСЛУГАМИ СВЯЗИ**

- *Новое видение архитектуры систем эксплуатации*
 - *Микросервисы и Платформенный подход*
 - *KDN – сеть определяемая знаниями*
- *Центр эксплуатации будущего – OpCF*
- *Системы OSS / BSS в архитектуре SDN / NFV*

Новая парадигма эксплуатации

Термины «Техническая эксплуатация сети связи», «Система поддержки эксплуатации OSS/BSS» относятся к тому времени, когда основным бизнесом поставщика услуг связи было создание и эксплуатация сложной телекоммуникационной сети. Хотя сети были сложными, услуги, предлагаемые клиентам, были относительно немногочисленными и простыми, и бизнес-модели аналогичными. Роль систем технической эксплуатации заключалась в том, чтобы сделать работу сети максимально эффективной и надежной, как правило, поддерживая группы специалистов, ответственные за отдельные аспекты эксплуатации.

На сегодняшнем цифровом рынке бизнес поставщика услуг связи намного сложнее и динамичнее, и поэтому задачи систем технической эксплуатации, систем OSS/BSS расширились. Возможно, сегодня следует подумать о роли OSS/BSS в качестве средства подключения бизнес-систем, клиентов и партнеров к базовой сетевой инфраструктуре, предоставляющей услуги.

ODA

Новое видение архитектуры OSS/BSS,
созданное ведущими операторами связи
(CSPs/communications service provider : AT&T, BT, Globe Telecom, Microsoft, Orange, Telefónica and Vodafone и др.)

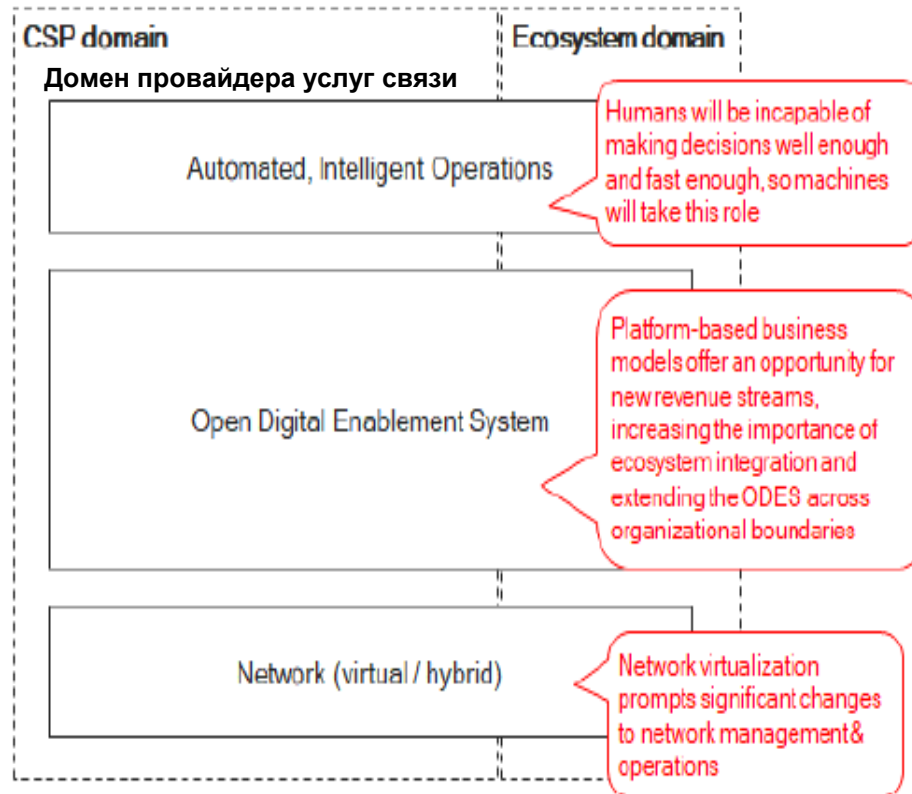
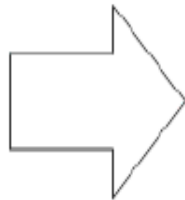
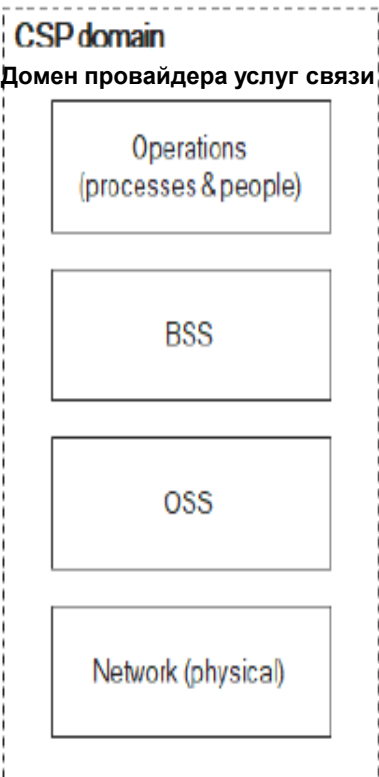
в рамках ТМ Форума для замены традиционных OSS и BSS, называется

Open Digital Architecture (ODA)
открытая цифровая архитектура
систем OSS / BSS

Драйверы для ODA

Traditional Telco Business

New Digital Service Provider Business



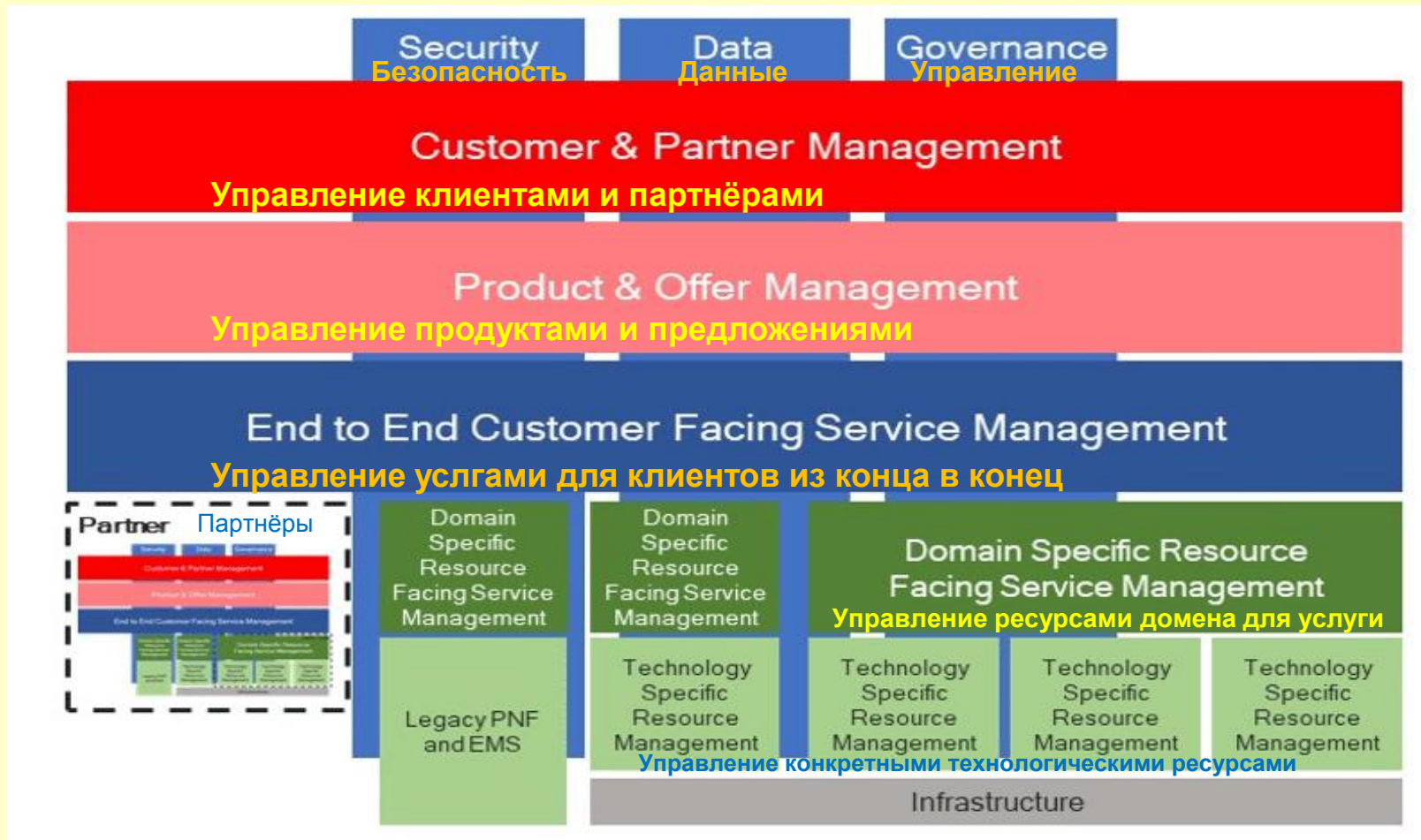
Люди неспособны принимать решения достаточно хорошо и быстро, поэтому машины будут выполнять эту роль

Платформенная бизнес-модель предоставляет возможность для новых потоков доходов, повышения важности интеграции экосистем и расширения ODA по всем границам организации

Виртуализация сети вызывает значительные изменения в управлении и эксплуатации сети

Source: TM Forum, 2017

Эволюционная модель – архитектура ODA



Принципы архитектуры ODA

- 1. Основана на стандартах, позволяющих создавать коммерческие и инновационные разработки с открытым исходным кодом (Linux)**
- 2. Объединены в единую архитектуру для OSS и BSS.** Эта архитектура будет предоставлять все возможности, ранее распределенные между независимыми BSS и OSS. Она устраняет барьеры между ними, создавая свободную связь между слоями, что повышает гибкость и снижает затраты, помогая CSP доставлять по-настоящему цифровой опыт работы с клиентами. Этот подход поддерживает требования будущих гибких, в режиме реального времени, инфраструктур провайдеров услуг, основанных на SDN и NFV
- 3. Ориентированная на данные архитектура с единой унифицированной плоскостью данных,** которая имеет основополагающее значение для автоматизации и искусственного интеллекта/artificial intelligence (ИИ/AI) и машинного обучения. Это означает, что используются согласованные и целостные данные на всех уровнях архитектуры. Реализует автоматическое управление на основе целей (намерений) из конца в конец, управляемых политиками.

Обеспечивает динамические сквозные процессы

Имеет общий репозиторий данных для всех слоев На первом этапе, скорее всего, увидим репозиторий общих данных на каждом из слоев архитектуры.

4. Базируется на Микросервисном компонентном подходе.

Так называемые *Микросервисы* с использованием **открытых интерфейсов API** от TM Forum для обеспечения гибкости построения:

- Строятся в соответствии с определением *микросервиса*, совместимым с TM Forum и позволяют контейнеризировать и поддерживать расширение возможностей через организационные границы в экосистемах. Используют рабочее имя «Framelets» для описания подкомпонентов архитектуры. «Framelets» - это Lego-подобные строительные блоки будущей архитектуры ODA. Нужно держать их на разумном уровне детализации: слишком мало, и мы увеличиваем сложность сборки, но слишком много, и у нас может не хватить гибкости.
- Framelets позволят поставщикам услуг получать архитектурные компоненты от разных вендоров, которые затем могут работать вместе, независимо от того, какая компания им предоставляет.
- Поставщики могут объединить несколько Framelets вместе, в одном продукте (при условии, что продукт предоставляет все **Open API** для данной комбинации Framelets). Наличие границ Framelet в архитектуре позволяет выбирать компоненты по принципу «plug and play» между ними, независимо от того, как они создаются. Определение границ Framelet будет регулярно пересматриваться: мерой хорошего компонента будет его повторное использование.

5. Способны к искусственному интеллекту (AI) и автономности

Простая автоматизация уже недостаточна, учитывая превышающие человеческие возможности скорости / сложности принятия решений. Поэтому цель будет заключаться в использовании управляемой событиями модели, основанной на целостном искусственном интеллекте, определяемом знаниями.

6. Способны работать в режиме реального времени (PPV)

- Все слои модели работают в режиме реального времени; это справедливо для слоя, обращенного к клиенту/экосистеме и для слоя, обращенного к ресурсам
- Ожидается, что большинство взаимодействий клиента с моделью будет через веб-сайт, при этом вся информация о клиенте, такая как использование, биллинг и согласование партнеров / экосистем, будет обновляться в PPV
- Взаимодействие с ресурсами также будет осуществляться в PPV, это означает, что инвентаризация и отчетность также должны быть в PPV
- Время разработки и время исполнения будут интегрированы во все уровни стека (при добавлении новых продуктов и услуг или даже новых сетевых конфигураций не потребуется «останавливать машину»), тем самым позволяя интегрировать рабочие бизнес-процессы и процессы эксплуатации и обеспечивать поддержку основанного на **DevOps** подхода
- Необходимо руководить изменениями ODA по мере разработки модели с политиками изменений, применяемыми для управления версиями Framelet.

7. Обладают способностями экосистемы, позволяющими легко взаимодействовать с партнерами.

8. Поддерживают платформенные модели и облачные возможности

- Позволяет делать открытыми/переносить Framelets (возможности актуализации платформы) в соответствии с платформами
- Позволяют интегрировать Framelets третьей стороны

9. Учитывают изменения в управлении и эксплуатации сети, вызванные виртуализацией сетевых функций (Network Function Virtualization/NFV).

DevOps



10. Ключом к функциональной архитектуре ODA являются слои (уровни) и абстракции, которые они представляют. Open API-интерфейсы ТМ Форума работают как интерфейсы между слоями, но API одного недостаточно. Для полного описания интерфейса требуется определение информации, на которую действует API, ожидаемое поведение и потоки процессов по обе стороны от интерфейса. Примером важности этого является то, как CSP с операционной деятельностью в нескольких странах, пожелают иметь некоторые слои общими (возможно, региональные или централизованные) и использовать локально адаптированные системы для других уровней (например, управление клиентами).

11. Фундаментальным требованием к концепции компонентизации и мультивендорности является то, что интеграция между компонентами должна быть через стандартизованные открытые API. Обеспечение того, чтобы одни и те же API-интерфейсы используются как для внутренней, так и для внешней интеграции, ускоряет создание партнерства внутри цифровых экосистем, при этом каждая сторона, выставляет возможности компонентов через открытые API.

Проект ODA строится и работает с программой Открытые интерфейсы API ТМ Форума, которая разрабатывает общий базовый набор основанных на REST открытых интерфейсах API, являющихся сервисами, что позволяет управление сервисами, даже если компоненты этого сервиса поставляются другими партнерами в экосистеме. Использование общего многократно используемого набор API для всех возможностей операционного управления значительно упрощает архитектуру.

12. Риски безопасности должны постоянно сопоставляться с бизнес возможностями в системах с открытой архитектурой, предназначенных для поддержки сложных бизнес-моделей, где организационные границы часто неоднозначные и услуги состоят из распределенных, отдельно поддерживаемых и разворачиваемых компонентов. Безопасность должна рассматриваться как на 10 уровне компонентов так и целиком с точки зрения всей системы.

Платформенный подход

Стратегия платформы имеет два ключевых элемента:

- **Платформенная бизнес-модель**, которая устанавливает цифровые экосистемы или рынки, соединяя потребителей с производителями товаров и/или услуг; и которая быстро эволюционирует в ответ на рыночные условия
- **ИТ архитектуры на основе платформы**, которые поддерживают электронный рынок и цифровые бизнес-модели, позволяя улучшить эксплуатационную гибкость.

Операторы могут использовать платформенный подход для преобразования всех аспектов их деятельности, от бизнес-процессов до систем эксплуатационной и бизнес поддержки (OSS/BSS)

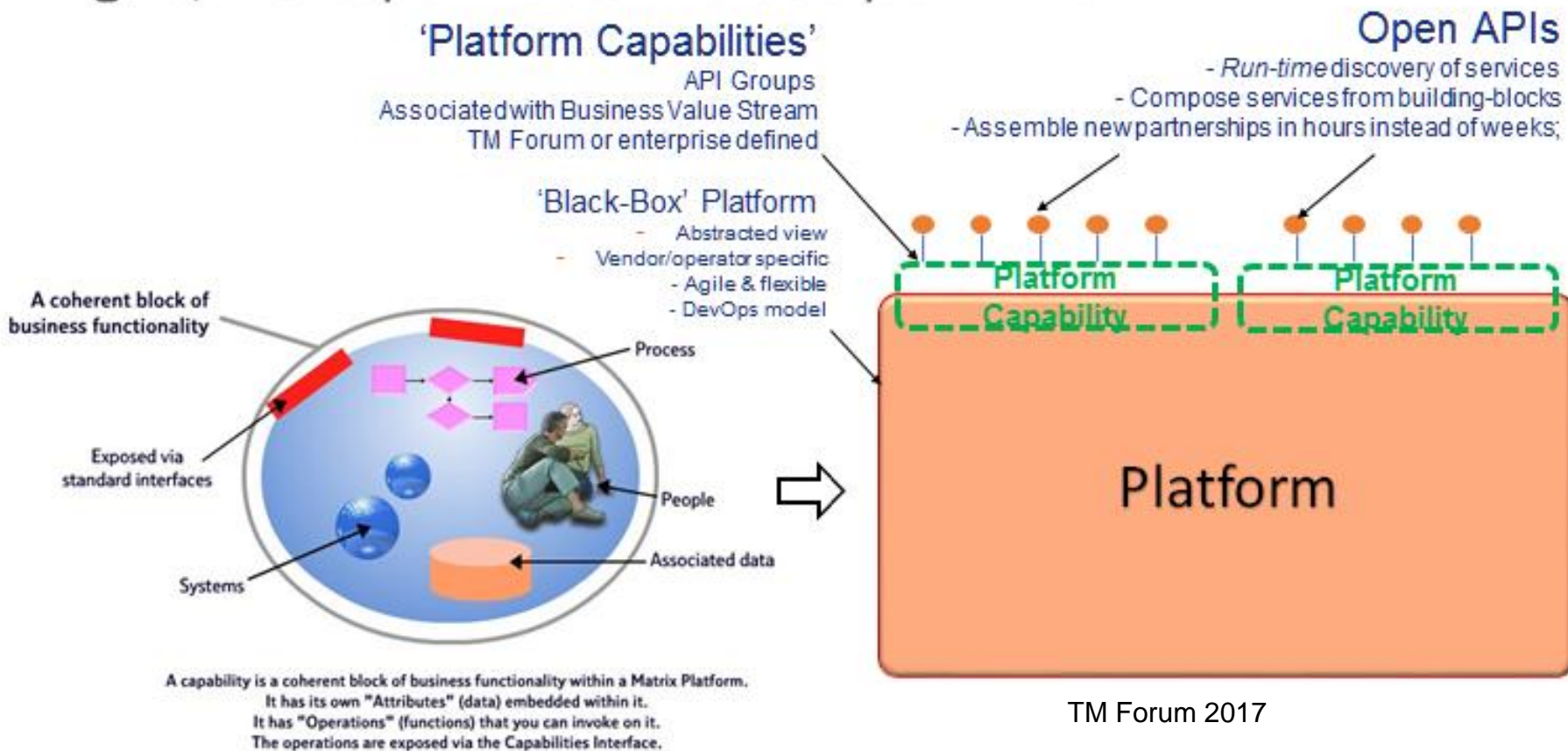
Ключевая концепция **платформы** заключается в том, что она является границей управления, которая инкапсулирует системы, людей/организации, процессы и информацию, управляемых как единое целое для предоставления определенного набора **возможностей для бизнеса** с помощью открытых интерфейсов ***Open API***.

Платформы предоставляют свои возможности через интерфейсы называемые **Platform Capabilities**, включающие в себя наборы API и правила их использования, которые могут определяться самими организациями

Составляющие платформы

Platforms

Agile, DevOps Platforms & Open APIs



Левая часть рисунка показывает, что платформа - это граница управления, которая предоставляет через стандартные интерфейсы один или несколько связанных блоков бизнес-функций - «Что», которые реализуются с использованием комбинации систем, людей, процессов и информации - «Как».

Возможность - это связанный блок бизнес-функциональности в пределах Матрицы платформы. Она имеет свои собственные «Атрибуты» (данные), встроенные в неё. Она имеет «Операции» (функции), которые можно вызвать в ней. Операции отображаются через Интерфейс Возможностей.

Искусственный интеллект (ИИ / AI) в OSS/BSS

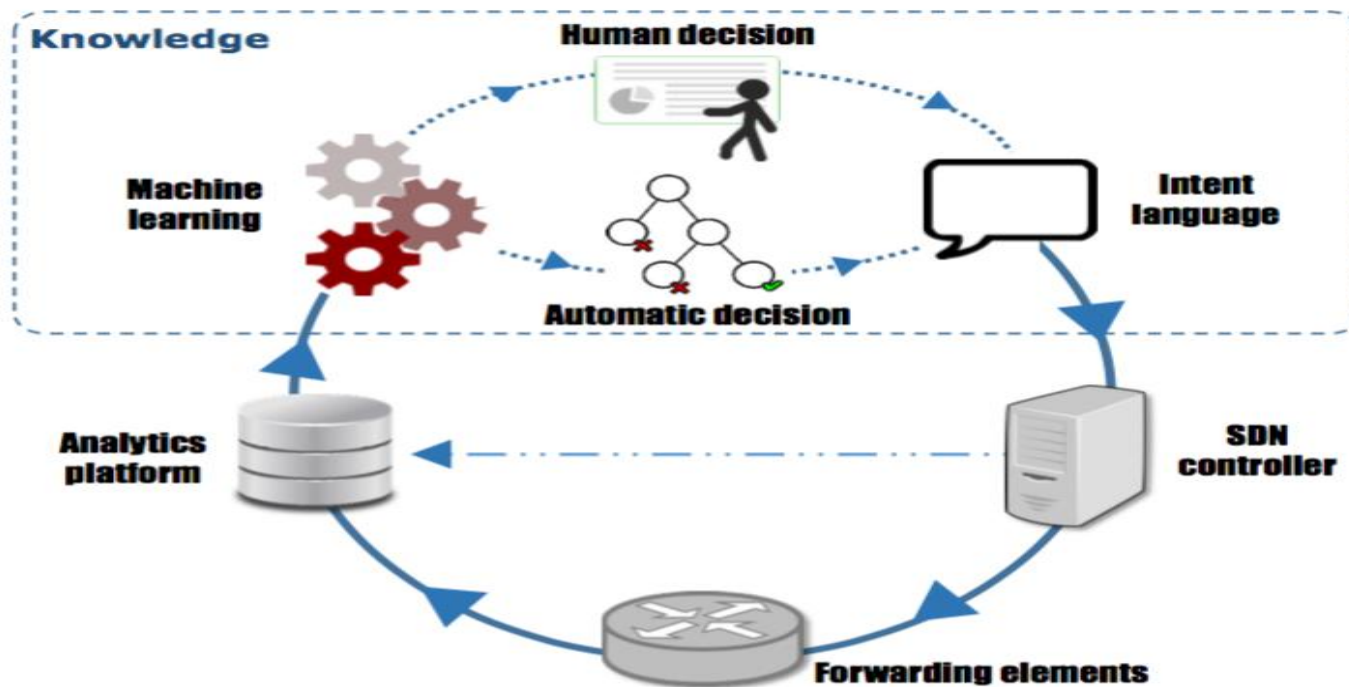
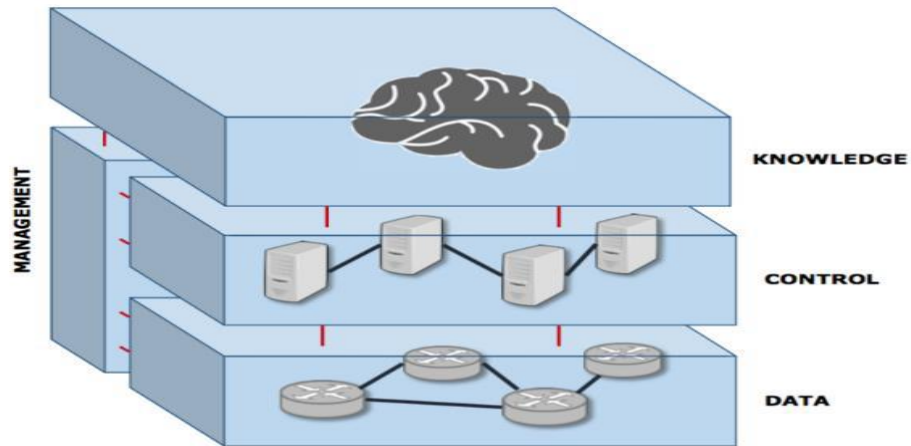
Системам OSS / BSS необходима гибкость (agility).

Например, традиционная инвентаризация сети относительно статична, но в инфраструктуре SDN и NFV виртуальные машины, предоставляющие услуги связи, могут быть сконфигурированы за считанные секунды в любом месте инфраструктуры. Это означает, что конкретный экземпляр клиента может определяться только с помощью решения по управлению ресурсами в реальном времени, которое постоянно обновляется. **Без более гибких OSS/BSS (ИТ) систем уровень SDN и NFV не может быть реализован.** Необходима автоматизация с замкнутым циклом для решения в режиме реального времени задач конфигурирования, управления услугами и оркестровкой. Эта задача в новой, высоко динамичной инфраструктуре становится ключевой .

Причем, просто автоматизации уже не достаточно: требуется ***интеллектуальная автоматизация*** комплексных решений на сверхчеловеческих скоростях. Для этого необходимо добавить искусственный интеллект (AI) для управления и эксплуатации сетей связи будущего.

На вершине **открытой цифровой архитектуры ODA** добавляется новый архитектурный уровень - **Плоскость знаний - A knowledge-defined network - KDN**

Сеть определяемая знаниями - KDN



Сеть определяемая знаниями

A knowledge-defined network - KDN

Новый архитектурная плоскость, названная «Сеть определяемая знаниями», которая опирается на машинное обучение и когнитивные методы, предложена для эксплуатации сети. Одна из самых больших проблем при применении машинного обучения для управления и эксплуатации сети состоит в том, что сети это по своей природе распределенные системы, где каждый узел (например, коммутатор, маршрутизатор и т. д.) имеет только частичный взгляд на управление всей системой. Обучение узлов, которые могут только просматривать и действовать на небольшой части системы сложно, особенно если конечной целью является осуществление управления за пределами локального домена.

Новая тенденция, заключающаяся в логической централизации управления (через SDN), облегчит сложность обучения в распределенной среде. Стратегия TM Forum состоит в том, чтобы реализовать **Плоскость знаний** для сети в контексте нижележащей базовой инфраструктуры **SDN**.

Машинное обучение нуждается в **данных**. Это требует перехода от современной архитектуры на основе процессов к архитектуре, управляемой данными.

A knowledge-defined network (KDN)

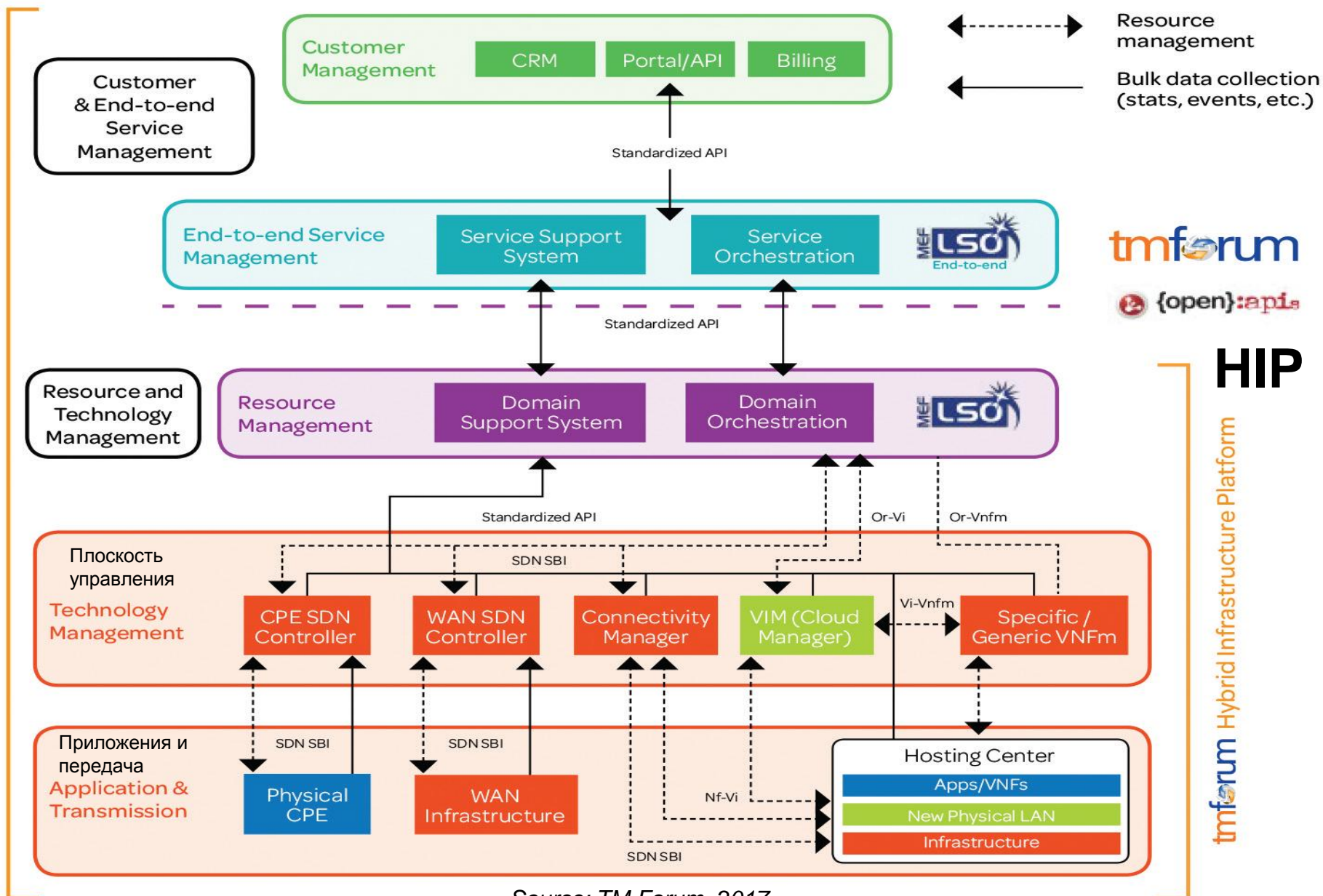
Сеть определяемая знаниями (KDN) работает на основе замкнутого контура управления для обеспечения автоматизации эксплуатационного управления, используя ИИ и машинное обучение в трех ключевых областях:

1. Управление опытом клиентов
2. Управление услугами
3. Автоматизация управления сетью



Например, операторы используют виртуальных агентов и чаты для улучшения качества обслуживания клиентов. Поскольку Internet of Everything берет в расчет и пользователей и сеть и управление сервисами, то невозможно без автоматизации поддерживать объем и скорость изменений, которые происходят в программно-определяемой сети, состоящей из миллионов узлов и тысяч приложений.

Структура Центра эксплуатации будущего OpCF



Source: TM Forum, 2017

Центр эксплуатации будущего OpSF

Open API – используются на разных уровнях в среде поставщика услуг связи: от управления клиентами до управления ресурсами и технологиями. Например, услуги могут быть заказаны через портал и затем автоматически активируются через гибридные сети, состоящие из виртуализированных и физических компонентов.

Возможно, самая высокая область активности в OSS сегодня находится на уровне сетевых услуг описываемой архитектуры. Сама сеть состоит из большого числа виртуализированных сетевых функций (VNF) . Сетевые услуги предлагаются путем динамической настройки и объединения этих функций вместе. Эта концепция часто описывается как Сеть как Услуга (**NaaS**), где сетевая услуга, которая ранее требовала физического соединения сетевых инфраструктурных устройств, теперь может **динамически создаваться и управляться на уровне виртуальных элементов**.

Рассматриваемая таким образом сеть может рассматриваться как тип платформенной архитектуры. Объединение виртуальных сетевых функций в такую платформу сетевых услуг, **абстрагирует и эффективно изолирует вышележащие слои** (приложения и пользователи) **от сложности отдельных VNF** (нижележащих технологий и сетевых уровней). Такое решение представляет собой эволюционное развитие OSS/BSS, обладающих последовательным набором интерфейсов для привнесения новых VNF в платформу, объединения их в сетевые услуги и последующего управления этими услугами.

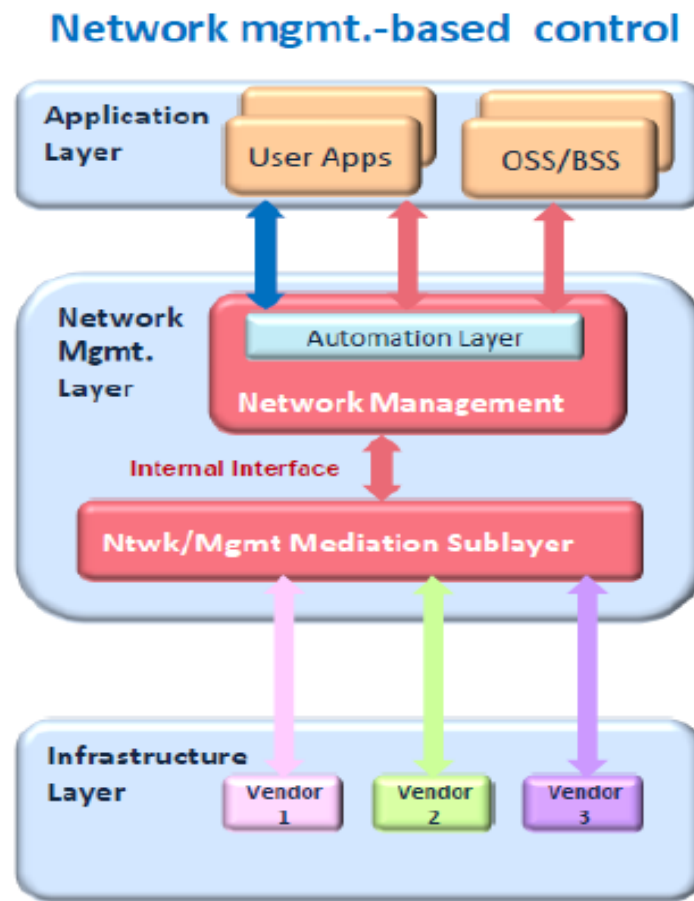
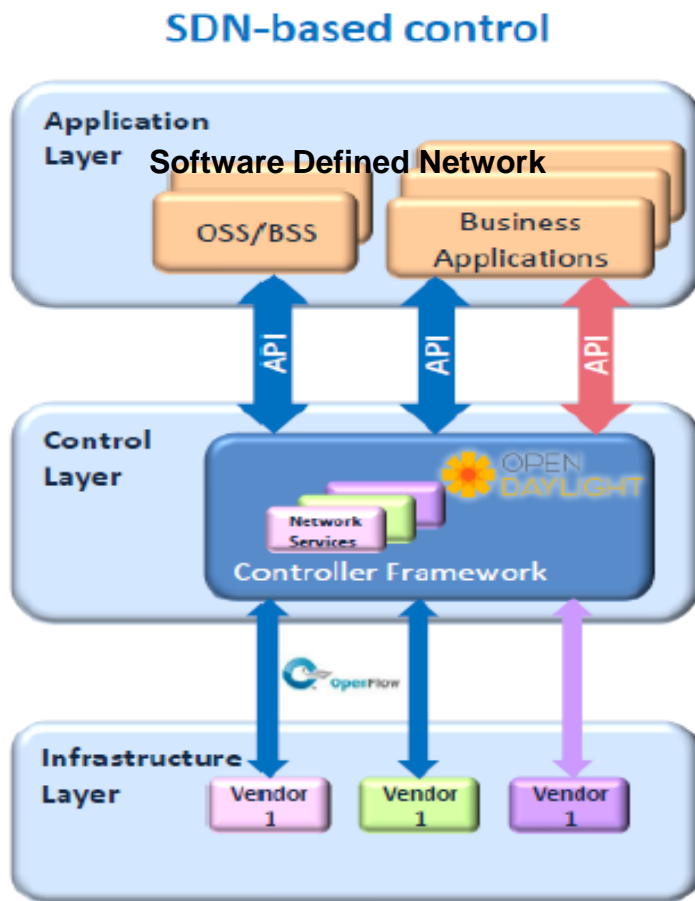
Интеграция SDN / NFV с OSS / BSS

Будет существовать разделение ответственности между традиционными OSS и вновь развернутыми контроллерами SDN и оркестраторами NFV.

OSS/BSS будет управлять относительно статичными параметрами конфигурации и ограниченными общими ресурсами, назначенными для подсетей или сервисов. Платформы контроллера SDN и оркестратора NFV затем динамически управляют этими сетевыми ресурсами для применения основанных на политике услуг в режиме реального времени для различных потоков трафика.

Системы OSS/BSS, соответствующие архитектуре ETSI NFV, должны поддерживать интерфейсы между традиционной OSS/BSS и NFV Management and Orchestration (MANO). OSS/BSS системы делегируют мелкомасштабное управление инфраструктурой NFV и специфическими VNF блокам VIM и VNF Manager, которые, в свою очередь, оркестрируются блоком NFV Orchestrator (NFVO). Таким образом, OSS/BSS будет отвечать за высокоуровневую конфигурацию инфраструктуры и сетевых функций, а структура NFV MANO будет управлять динамическими аспектами инфраструктуры и сервисов.

Системы OSS/BSS для SDN



С появлением программно-конфигурируемых сетей подход к эксплуатационному управлению меняется. Концепция единой точки управления сетью предоставляет системе OSS единый интерфейс мониторинга и сбора статистики, а разделение сети на плоскости дает возможность управлять каждой плоскостью в отдельности.

Системы управления сетью (NMS) конфигурируют сетевые элементы (NE) статически, при относительно медленном времени ответа, в то время как контроллер SDN будет работать в режиме реального времени.

Системы OSS для SDN и NFV должны учитывать *динамические* изменения в сети и предоставить возможность Контроллерам SDN и Оркестраторам NFV динамически применять изменения, зависящие от политики, в ответ на требования приложений или трафика при сохранении полной поддержки FCAPS.

Контрольные вопросы

1. Драйверы для ODES/ODA?
2. Эволюционная модель OSS / BSS – архитектура ODA, принципы?
3. Платформенный подход и составляющие платформы.
4. Искусственный интеллект в OSS/BSS. Определяемая знаниями сеть (KDN).
5. Центр эксплуатации будущего – OpCF – четыре особенности и структура.
6. Ключевые компоненты архитектуры SDN и взаимодействие с OSS.