SeaChange®

WHITE PAPER

A Universal Service Delivery Platform enables telco, cable, satellite, digital terrestrial and hybrid operators to deploy, brand and bill a rapidly expanding universe of content and services at lowest total cost of ownership.

Quad Play Requires a Universal Service Delivery Platform

8/2006

In recent years a lot of attention has focused on communication protocols as the key to quad play — the ability to blend content and programming across a branded offering that combines TV, phone, Internet, and wireless services. The conventional wisdom states that IMS (IP Multimedia Subsystem) must be fully deployed for consumers to enjoy the quad play experience. Once diverse access types (IPTV, DSL, WAN, GPRS, etc.) can interoperate then diverse services can converge.

But if the promise of any content — on any service — on any network — and on any device — is to happen, then the language of diverse services is not the only thing that must become universal. So must the service delivery platform on which services run.

In fact, the platform comes first — specifically the platform most in demand now: IPTV. IPTV is the next-generation service that is the most technology intensive and offers the most eye appeal to consumers — and it's fully deployable today. Place the right bet here and an operator enjoys both a short and a long-term advantage.

In the short term, operators can generate revenue from the on-demand options that consumers want now. In the long term, they can add even more quad play options as IMS arrives — without ripping out existing infrastructure, operating multiple technology stacks, or prematurely investing in technologies not yet ready for prime time.

The fact is, at some point *every* operator makes a platform decision about how to deploy services — quad play or otherwise — even if by default. That decision either ignores the difference between a service and the delivery platform on which it runs — or exploits the difference to increase scalability, flexibility, performance, and other benefits.

To ignore the difference means that every service brings its own client, its own navigation, its own way to capture user preferences, its own business rules engine, and so forth. To exploit the difference means that diverse services can share common functions within a middleware layer. Not only does this reduce duplication and operations costs, it allows operators to plug and play best-of-breed functions — like digital rights management, set top boxes, and video servers — into services where that makes logical sense. Even where operators don't want to *replace* functions, they may still wish that functions be *decoupled*— for example, user entitlement management from content asset management — creating new opportunities for market differentiation.

Nowhere are these opportunities more evident than in a multi-business operator model. In this model, larger operators sell services to smaller operators who resell those services under their own brands to consumers. This model works only if the services are truly differentiated. Operators cannot simply resell the same programing packages, the same business rules, the same look and feel — in other words, the same *brand experience* — as competitors. But if functions are decoupled, they don't have to. How they tailor one function does not limit how they tailor others. That gives them and their customers what they demand most, both in IPTV and in quad play — ultimate choice.

What Consumers Demand — Ultimate Choice

Because choice is so important to the consumer, it sets the ground rules for the operator, and therefore the technology.

Consumers want choice of both *content* (movies, games, music, etc.) and *context*. That's content delivered when they want, where they want, and in the form they want — PVR, DVR, VOD, or SVOD — live, prerecorded, on-demand, or interactive. They want their TV on the web, the web on their TV, videos on their wireless phones, and email and text messaging on everything. They also want to set content and context preferences in whatever way works for them — for example, via a set top box, a PC, or a mobile phone. In addition, different members of the same household will express different preferences.

Helping consumers control context and content is a great example of where operators can make money from a platform-enabled quad play lifestyle — provided the platform has features like a:

- **Preference Engine** — for setting content and context
- **Recommendation Engine** — for alerting consumers to content and context preferences they might enjoy based on previous choices
- **Search Engine** — for finding content from "the long tail," i.e., all the movies, home videos, photos, network programs, etc., available regardless of location

Some consumers may in fact be satisfied to watch only scheduled network programs on an analog TV set and never wish to pause the action to check email. What counts is the freedom to indulge as many preferences as possible in ways that create meaningful brand differences.

What Operators Need — A Difference

If choice defines the consumer, difference defines the operator. Difference is how you stand out, how you avoid competing on price, and how you brand. It is a reflection of the choices you offer your target consumer — for example, either by marketing a greater abundance of choices in more accessible ways or by marketing a more targeted set of choices to a specific demographic.

CONSUMERS DEMAND:	OPERATORS NEED:
• Content Whenever & Wherever	• Reliable On-demand System
• Flexibility to Tailor and Manage Services on their own	• Rich Portfolio of On-demand Applications
• More Interactive Experiences (PVR/DVR)	• On-demand Application Deployment
• More On-demand Experiences (VOD/SVOD)	• Low Costs
• Affordable Prices	• State-of-the Art Functionality and Performance
• State-of-the-art Function and Performance (HDTV)	

Table 1: Consumer demands drive operator needs

Either strategy creates differentiation from core product attributes — those outlined on the right side of Table 1. But sustaining those attributes— keeping applications exciting at affordable cost over time — is a challenge. What makes an offer different today will not make it different tomorrow. To sustain product differentiation over the long run operators need four key success factors working in their favor:

1. Speed to Market

Operators need to bring new service options to the consumer at least as fast as competitors do. The challenge: Unlike in previous eras, where services were functional "stovepipes," services today cannot be considered deployed until they are blended with other services — which can take time. Furthermore this blending occurs at multiple points — for example, in the collection of billing data and in the presentation of client navigation. Users will not tolerate the equivalent of yet another "TV remote on the coffee table" every time something new is added to the service portfolio. Marketers meanwhile will want the opportunity to invent new products that combine features of various quad play services in interesting and creative ways.

2. Revenue Growth with Reduced Customer Churn

Top line growth requires keeping existing customers happy while attracting new ones. At times, these two objectives may seem in opposition. Existing customers expect continuity with what attracted them to the operator in the first place while both existing and new customers are attracted by new offers that are better than what they have now. The optimum solution is to build on what works today — like video on demand and games on demand — in a way that is easily extensible tomorrow.

3. Low Deployment Risk

Deployments grow and become more complex as new services, applications, product configurations, and customers are added. Growth and complexity increase risk. There are more things to go wrong and ways for things to go wrong. In addition, as services become less stovepiped and more interdependent, a problem in one place is more likely to cascade to other places. *Scalability of the solution is key.* Deployments that start small should be able to grow quickly yet remain stable as operators gain customers and experience. You increase scalability when you componentized functions as discrete modules with clean, well-planned interfaces that act like fire doors that close when bad data tries to pass through. Isolating discrete functions also makes it easier to pinpoint an issue — or swap out a faulty component without impacting working parts of the system.

4. Low Total Cost of Ownership

"Our costs are less so we can sell for less" — a refrain often heard at auto dealerships — also applies to quad play operators. The lower the costs, the more attractive the prices, and the more money left over to deploy new services. Lowering deployment risk reduces cost of ownership — another reason to implement easily extensible modular systems. Modularity also allows operators to swap in best-of-breed components — components that may be less expensive to buy and more cost-efficient to use. The set top box (STB) is a great example, accounting for 70-80% of operator deployment costs — with most of that coming from proprietary client functionality buried within silicon and C code. These costs can drop substantially to the extent that:

- STBs are interchangeable commodities
- Clients are written in a portable language like Java, JavaScript, or HTML that can be easily modified
- Performance-optimized functions are server-based

Another savings opportunity is the effort needed to integrate services with an operator's existing BSS/OSS (business support system/operator support system). The more seamless the integration, the faster and less costly a service is to implement and maintain over time. And to the extent any service is able to inherit these types of cost savings, the greater the opportunity to keep offering competitive services at competitive prices.

It's All About the Architecture

Although these are four different success factors — speed to market, top line growth, low deployment risk, and low total cost of ownership — they have a lot in common, a fact that points to a single underlying strength. Here's what they have in common:

Operator Success Factors are Mutually Reinforcing. It is no accident that an operator that knows how to speed new services to market will also likely experience healthy top-line growth, or that an operator with low deployment risk will enjoy low cost of ownership. Risk increases cost. And slow deployments will likely turn off both existing consumers and miss windows of opportunity open for attracting new ones.

Operator Success Factors Exploit Common Technology Features. Success factors are mutually reinforcing to the extent common technical attributes speed services to market (like the ability to plug and play functions easily) and *also* reduce risk and lower cost of ownership — which again spurs top line growth either by attracting new business or reducing customer churn.

Operator Success Factors are Architecture-dependent. What makes these similar features alike is architecture — *i.e., a conscious strategy to map components into services based on cost, performance, and best-of-breed functionality.* For example, to deploy the user interface of an electronic program guide as a discrete function allows users to swap out commodity STBs that cost less and perform better.

Operator Success Factors are Service-agnostic. Whether the service is IPTV, mobile telephony, web browsing, or some combination — *the distinct value of architectural advantages is that they tend to apply everywhere.* They can bear fruit again and again over different services and over time. Take the example of a modular BSS/OSS adaptor.

Once an adaptor module exists for easy integration with a specific BSS/OSS, it does not have to be reinvented for every service or application. That speeds deployment, reduces risk, and reduces total cost of ownership all at the same time — and for *any* service or application that needs to talk to the BSS/OSS — all of which attracts new customers and helps satisfies existing ones with new offers.

Convergence on Trial

What this shows is that architecture, done right, provides a powerful single point of leverage for keeping an operator's offering fresh across services, applications, and content. Creating brand differences, however, takes more than just a fast, low-cost, low-risk, and sustainable way to refresh services. You also have to tailor services as differentiating user experiences that stand out in the consumer's mind. That's why some device makers are doing limited rollouts of convergent services (e.g., programmable video recording via mobile). They're hyping these trials as harbingers of more ambitious hybrids to come. But operators aren't where device makers want them to be. Rather than architect deployments around a device, operators would rather deploy devices around a deployment architecture — so they can entertain *all* branding possibilities.

As Table 2 shows, operators have an almost unlimited number of ways to tailor their offerings. Consider that there are four basic services that define quad play; each of which consists of a long list of elements — and each of those can be tailored differently depending on the service, user preferences, cost, and other factors. Navigating a cell phone menu will be different in some respects (and the same in others) as navigating a TV's electronic program guide.

	SERVICES			
ELEMENTS	TV	Phone	Web	Mobile
Content	• On-demand • Interactive • Convergent	• Interactive • Convergent	• On-demand • Interactive • Convergent	• On-demand • Interactive • Convergent
Applications	• On-demand • Interactive • Convergent	• Interactive • Convergent	• On-demand • Interactive • Convergent	• On-demand • Interactive • Convergent
Hardware	• On-demand • Interactive • Convergent	• Interactive • Convergent	• On-demand • Interactive • Convergent	• On-demand • Interactive • Convergent
Business Rules	• On-demand • Interactive • Convergent	• Interactive • Convergent	• On-demand • Interactive • Convergent	• On-demand • Interactive • Convergent
Navigation	• On-demand • Interactive • Convergent	• Interactive • Convergent	• On-demand • Interactive • Convergent	• On-demand • Interactive • Convergent
Entitlement	• On-demand • Interactive • Convergent	• Interactive • Convergent	• On-demand • Interactive • Convergent	• On-demand • Interactive • Convergent
Promotions	• On-demand • Interactive • Convergent	• Interactive • Convergent	• On-demand • Interactive • Convergent	• On-demand • Interactive • Convergent
Other...	• On-demand • Interactive • Convergent	• Interactive • Convergent	• On-demand • Interactive • Convergent	• On-demand • Interactive • Convergent

On-demand Examples:
VOD, Games On Demand, Broadcast

Interactive Examples:
Red Button Apps, Voting, Ecommerce

Convergent Examples:
Caller ID, Video Blogging, Home Media Programing, PVR via Mobile

Table 2: When tailoring an offer, an operator should entertain all combinations of services, service elements, and service types (on-demand, interactive or convergent).

Good branding would call for elements like color, company logos and page layout to be consistent. But content sources would probably be different — and presentation style could also change depending on whether the presented information were on-demand TV listings, unread emails, or something else.

Content branding calls for access to specialized services and tools, such as those that help:

- Design EPGs
- Add logos, colors, and background to set top clients
- Design backgrounds, slideshows, wall of posters and other user interfaces
- Acquire and aggregate content
- Compile and promote service bundles
- Create MPEG assets with multiple video backgrounds and still frames, video and audio elements

Non-content branding elements would include entitlements and business rules, functions that determine, respectively, which content or applications a user is entitled to consume and under what conditions. Entitlements may apply one way when the content is a premium on-demand movie the user has rented — and which the business rules say can be viewed up to a certain number of hours after payment. But entitlements and business rules may apply a different way when the content is a music video the user has purchased for download off a music service.

Added to this mix of service-versus-element is another layer of complexity. That is: service-versus-element-versus *type* of service — i.e., on-demand, interactive, and convergent. In a convergent system, for example, on-demand movies might be played on either a HDTV home theater or on a wireless phone — among other possibilities. Again, various entitlements, business rules and other elements will apply differently depending on which is the case. Furthermore, if the movie has interactive content — e.g., alternate endings, premium tie-ins to a website, etc. — then additional entitlements and business rules may also apply. For example, you may be able to chat with a star in the movie or insert your own image in a scene.

What's clear from this picture is that there is far more to keeping your offering fresh than just updating the various elements of content, applications, entitlements, and so forth. You also have to tailor these elements in synch with audience demands while reinforcing a clear brand distinction. As Table 2 shows on the previous page, there is an almost unlimited number of ways to do that — especially when convergent (IMS-enabled) applications become available.

Get Ready for IMS Now

In fact, when IMS-enabled applications do become more available, operators will jump into the tailoring game or quickly become commoditized. That's because IMS only has value in a blended environment — and to blend applications you have to tailor them. After all, what is the point of allowing applications that are running on different services to blend if they do not in fact blend? And if they do blend, then there must be value propositions to support that effort. As these value propositions emerge, they will drive further blending, further tailoring, and even more value propositions, which will drive further tailoring and blending — and so on and so on.

This begs the question: Why wait for an IMS architecture to emerge? As Table 2 shows on the previous page, there are a lot of mixing and matching opportunities that can be exploited today to differentiate service offerings. Why not jump in now and get a head start at building the expertise and infrastructure required to do that? Operators cannot make money selling convergent applications that don't yet exist. But they can make money selling a convergent experience. That's an experience where applications:

- Are consistently branded across services
- Invoke elements tailored to satisfy consumer choice and create brand differences
- Are highly interactive
- Satisfy on-demand requirements

Selling a convergent experience before IMS actually takes hold offers three big advantages: 1) It gives the operator credibility in the marketplace as a forward-thinking provider; 2) It gives the operator experience at running convergent applications so the transition to IMS will be smoother; and 3) It implies you already have in place a framework for efficient quad play deployment — where discrete service modules share a componentized backend infrastructure you don't have reengineer every time consumers want something new.

IPTV: Giving Consumers What they Want

In fact, the only reason *not* to jump into the tailoring game now is if you *did* have to reengineer your infrastructure every time consumers want something new. If your billing function only works with IPTV and not with mobile, it will be hard to add mobility to your IPTV offering. If you can't stream TV to both a set top box and a wireless PDA, that's another blended offering you can't provide. When what you own is a collection of stovepipe applications, then the *only* way keep ownership costs down is to *not* reengineer them. The problem is — what happens when a competitor in your market area can do both: mix and match service elements *and* keep total cost of ownership down?

But — an operator might argue — what happens when IMS technology does arrive and convergent deployments do become widespread? Won't we have to upgrade then anyway? Won't that level the playing field? In other words: Does *any* infrastructure investment make sense now, given the risks of how future technologies might evolve?

But here is another question: What about IPTV? IPTV is a technology that is fully deployable now and is certain to be at the center of any future IMS-based offering. It is also the technology that most engages consumers *today*.
IPTV already offers:

- High definition digital quality video
- Surround sound stereo
- Mass audience entertainment content

- On-demand delivery
- High interactivity
- Digital lifestyle features like:
  - »Stop live action
  - »Time shifting
  - »Electronic program guides
  - »Program-in-program viewing
  - »Camera selection

These features are likely to form the center of gravity for any new quad play investment — pre- and post-IMS — precisely because they are the most sensory engaging and therefore also the most technology intensive. Also, IPTV demonstrates an important lesson about quad play: users, and therefore the services they want, often drive the timing of infrastructure investments. Witness the demand for expensive plasma screens, HDTV receivers, and most recently Blu-ray Disc™ DVD players. If operators are willing to wait until IMS standards completely sort out before they experience the IPTV difference, users are not.

That means to get quad play right you better get IPTV right first. If you are a cable operator, then IP video services are the base from which you will move into other services. As you go, you want to avoid ripping out existing infrastructure or operating multiple technology stacks. If you are a telco, you want to avoid that fate too — but in your case rather than build *off* an existing IP video base, your task is to *add* IPTV to existing telephony and wireless. So in either scenario — building out or adding on — it's your choice of IPTV solution that will largely decide how "IMS ready" your offerings are "out of the box." In other words, can you easily plug new services into existing ones and can you easily tailor and blend the services you already have?

But telcos have a greater stake in their IPTV decision than cable operators. That's because consumers already think of cable companies for TV — and because telcos with no TV customers have no TV consumers. Every telco IPTV customer must at some point be taken away from an entrenched TV competitor. Telcos must also ask consumers to change how they think about where to get TV — a much harder task than just winning consumers from other telcos.

That means the first time telcos offer their customers a convergent experience it better be great. And they have to do it without making mistakes, especially when it's still early — before they've had a chance to earn a good reputation. As the cable industry has discovered, consumers can be very unforgiving when it comes to their TV service.

IPTV as Universal Service Delivery Platform

So whether you are a cable provider or a telco, IPTV plays two roles — to deliver a branded TV offering that also anticipates the quad play experience of IMS. That is to say:

IPTV factors for quad play success	**Will tend to be**	**And architecture-enabled by**
• Speed to market • Revenue growth with reduced customer churn • Low deployment risk • Low total cost of ownership	• Mutually reinforcing • Based on common technology features • Architecture-dependent • Service-agnostic	• Functions componentized as discrete modules • Cooperation between modules via clean, well-planned interfaces with built-in safety features • Reusable adaptors for easy integration with external services like BSS/OSS • Components deployed based on cost or performance

This describes a *Service Oriented Architecture (SOA)* in that services are decoupled from each other as discrete modular components. In particular, back office functions (like those in Table 3 on the next page) are decoupled from the end-user applications (like EPG) they support. That makes all services easier to isolate, and therefore tailor, than when functions are hardwired into an entire service.

Major Function	Functionality
Subscriber/Account Management	Device management, service allocation, provisioning, billing, preference and profiles, transaction history
Product Offering/Campaign Management	Packaging, pricing, rating, promotions and discounts
Digital Rights Management	Content, operator, and subscriber protection
Service and Asset Management	Channel lineup, preferences, pay per view management, service configuration, barker channel, application service management
Network and Systems Management	Video service assurance, alarms and monitoring, performance/ video monitoring, administrative dashboards, ticketing and troubleshooting, content lifecycle management

Table 3. Back office functions are decoupled but shared in common by quad play applications in a services oriented architecture

SERVICE ORIENTED ARCHITECTURE: KEY FEATURES

N-tier
Set top box / service middleware / operator functions

Functional partitioning
Navigation / entitlements / asset management / etc.

Performance optimization
Performance-optimized streaming hardware / network-based QoS / etc.

Application decoupling
Open interfaces / standard protocols, etc.

Ready-to-market applications
Video on-demand, TV mail, etc.

Elements are also easier to swap out — like a set top box or a digital rights management module — where doing so creates performance or functional leadership. And any cost or performance benefits — derived either by tweaking a particular module or replacing it entirely — are inherited by all services that use that module. So rather than increase total cost of ownership or deployment risk, tailoring an element like business rules — say, to work one way for movies and another way for games — can actually *decrease* total cost of ownership and deployment risk. Then when you go beyond IPTV — say, from movies and games to instant messaging — you still only have one business rules engine to deal with. And it's the same rules engine with which your technicians are already familiar — which again drives down total cost of ownership, deployment risk, and time to market.

Five Common IPTV Architecture Mistakes

These examples show how IPTV, properly architected, facilitates quad play deployment — *exactly the type of deployment IMS anticipates.* But the reverse is also true. Architect IPTV the wrong way and it becomes a huge quad play (and therefore IMS) roadblock. Consider, for example, five common IPTV architecture mistakes: end-to-end, application centric, organic, device centric, and BSS/OSS centric.

1. End-to-End

End-to-end deployments view IPTV as essentially one closed monolithic system. Everything is hardwired to everything else—so changing a function here has the potential of breaking a function there. Also, functions are designed to work specifically with each other — i.e., they are *tightly coupled.* That means that all functions are only designed to work in a single context, for example, on-demand video. Making the same function (e.g., navigation) support a different context, (e.g., video downloads to wireless phones) might not be possible. Finally, both operators and users are locked into using vendor-supplied functions. All of the above make solutions hard to tailor or scale without compromising total cost of ownership and other quad play success factors.

2. Application Centric

This is when the entire IPTV offering is deployed as a software layer that rides on top of operating systems, middleware network services, and the hardware. Application centric architectures also tend to be end-to-end, with the single application trying to do everything. (As opposed to a Service Oriented Architecture (SOA) where application functions are deployed as discrete best-of-breed components.) Typical application centric examples come from PC software companies looking to move into the TV space. As such, they often lack the "industrial strength" expected of a true IPTV company with deep expertise and experience across the entire IPTV technology stack.

3. Organic

Also called homegrown, this is where an operator builds out its own service architecture by stitching together services and functions as these are added to the deployment. The problem with organic solutions is that they are designed around the needs of one particular operator. They are not designed to transfer well to other operators. As a result, the alignment of functions versus modules may be inconsistent and unclear — interfaces may be ad hoc — and a lot of overlap may exist between modules. As with end-to-end, organic solutions don't scale well and tailoring is impossible without undermining quad play success factors.

4. Device centric

This architecture is also very organic, except that now it is the device vendor (say, a set top box maker) that stitches the modules together, rather than the operator. The architecture suffers from the same weaknesses as organic. It also suffers from the additional weakness of being oriented around a particular component of the solution — i.e., the vendor's device — rather than the solution as a whole.

5. BSS/OSS centric

A BSS/OSS is the infrastructure that enables telcos to create, deploy, manage, and maintain network-based services. A BSS/OSS-centric services architecture, however, is something of an oxymoron since the BSS/OSS itself is not fundamentally concerned with services. It's not designed to know about entitlements, business rules, client navigation, and the other shared service resources. It *is* designed to know about infrastructure resources like QoS, switch configurations, firewalls, and dial tone. Organizing shared service resources within a service oriented architecture — on top of the BSS/OSS — avoids trading network priorities off against service priorities and vice versa. It also avoids the painful, time consuming, and potentially destabilizing effects of reengineering the infrastructure every time you want to add or tailor a service.

Deploying a Service Oriented Architecture

Organizing around services — IPTV first — and then convergent quad play — is a layered process. The strategy is to pick the low-hanging IPTV fruit first. Give consumers a convergent experience they will enjoy now, but also nurture quad play success factors as IMS builds out. That strategy relies on an SOA that decouples functions (that are simultaneously service-enabling and networkagnostic) as discrete functions — decoupled from each other and also from the BSS/OSS that controls the larger infrastructure on top of which they run.

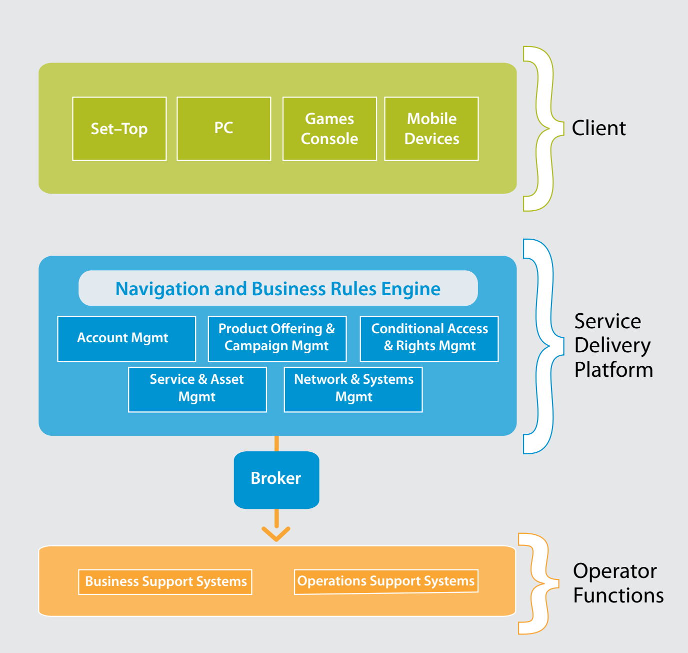

Set–Top
PC
Games Console
Mobile Devices
Client
Navigation and Business Rules Engine
Account Mgmt
Product Offering & Campaign Mgmt
Conditional Access & Rights Mgmt
Service & Asset Mgmt
Network & Systems Mgmt
Service Delivery Platform
Broker
Business Support Systems
Operations Support Systems
Operator Functions

Figure 1: Services Oriented Architecture

Those functions include: business rules, set top boxes, clients, navigation, electronic program guides, entitlement, content asset management, ordering, and billing. Actually, the specific functions are not what are important since the whole point is to be able to plug in new functions (like those supporting new quad play applications) which might not have been anticipated when IPTV was installed.

Figure 1 shows this architecture in action as a universal service delivery platform. There are three layers:

Client

This environment is where the user interacts with the world of on-demand, interactive, and convergent applications — everything from movies on demand to games to TV mail. As such, it is operating system, STB, and network agnostic. Core technologies include third-party application launch and connectivity with back office functions. To shed costs, clients may be deployed on devices that are both resource-constrained and sourced from multiple vendors providing diverse services supporting open standards such as Java, JavaScript, and HTML, and emerging standards like OCAP and MHP.

Back Office

This is where application logic resides — both the logic shared by multiple services (like billing or business rules) and application-specific logic (like TV mail). Some functions (like navigation) talk to client code. Some (like business rules) don't talk to a client but do talk to other functions (like conditional access and rights management). Some back office functions also talk to BSS/OSS Integration functions in layer three. For example, billing middleware that tracks movie rentals will talk to the billing system that actually invoices the customer.

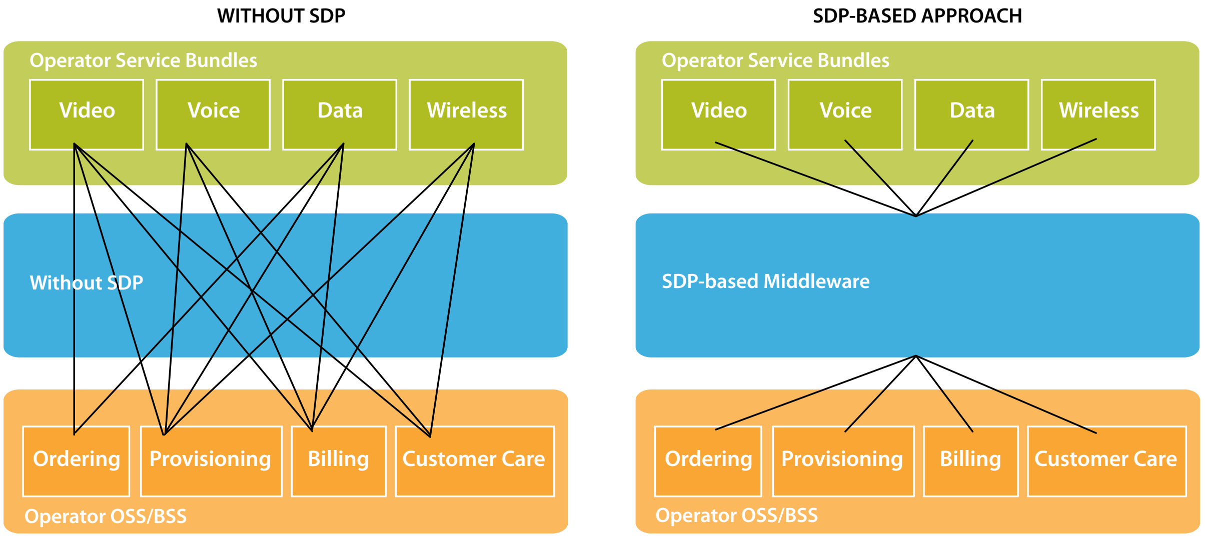
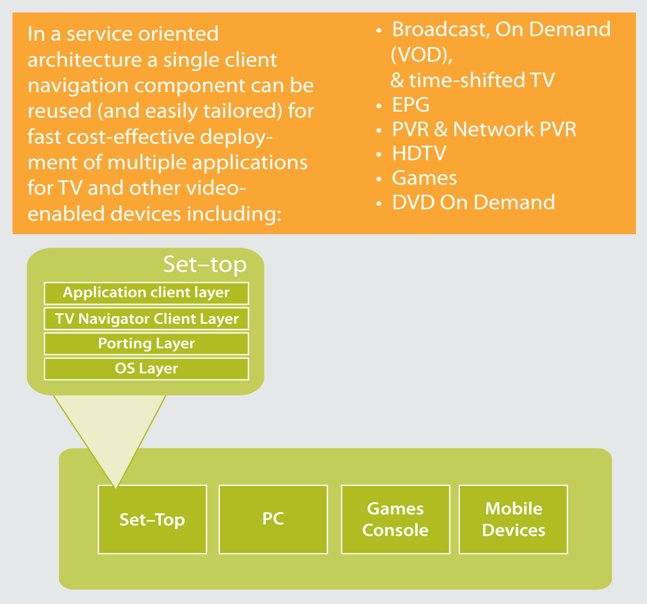


Figure 2: An SDP vs. A Non SDP Approach

BSS/OSS Integration Layer

This layer provides invoicing, customer care, legacy data access — and other services external to the quad play technology stack — that support administration of the business rather than actual service delivery.

To get an idea of how this approach streamlines operations and deployment, look at Figure 2. In conventional deployments (left side), functions are not componentized, and components are not decoupled. Every piece of logic is hardwired to every other piece of logic that it needs to create an application. This creates a "rats nest" of one-to-one connections that make tailoring and updating virtually impossible while still providing low total cost of ownership and other success factors.



In a service oriented architecture a single client navigation component can be reused (and easily tailored) for fast cost-effective deployment of multiple applications for TV and other video-enabled devices including:

- Broadcast, On Demand (VOD), & time-shifted TV
- EPG
- PVR & Network PVR
- HDTV
- Games
- DVD On Demand

Figure 3: Navigation as Deployed within the Client Tier of a Services Oriented Architecture

In a service-oriented deployment (right side) functions are componentized, and components are decoupled. This allows services to easily exploit common facilities without duplication or a lot of overhead (human or technology). On the client side — this allows fast, incremental deployment of best-of-breed applications into commodity hardware (see Figure 3). In the middleware layer — this allows creation and deliver of revenue generating video and other quad play bundles. And on the operational layer — this allows full integration with BSS/OSS, customer support, accounting, and other backend systems through an open, flexible, scalable architecture — without duplicating or replacing the functionality of those systems.

A service oriented architecture also enables easy transition to an IMS world. It allows service providers to gain experience with the type of blended distributed environment IMS envisions. Operators reuse functions such as device management and content delivery across applications now just as they will reuse them later with IMS — providing a high level of service continuity in the transition process. This is the sweet spot of IPTV deployment: a great suite of IPTV applications and content within a plug-and-play architecture that accommodates choice, tailoring, and performance, keeps cost of ownership low, and protects your investment heading into the IMS future.

The Ultimate Test — Multi-Business Quad Play

Any operator can exploit the benefits of a service oriented architecture on which to deploy a universal service delivery platform. That's because services are no longer defined by particular end-to-end collections of logic and rule sets. However you wish to define your business — whether by introducing new services and applications or by creative tailoring of the services and applications you already have — is up to you. Navigation, entitlements, business rules, content asset management, access control, and other core middleware components are now decoupled — so you are now free to use, and reuse, them differently for different applications and different kinds of content.

These advantages benefit one operator business model in particular, however. That's the multi-business operator. In this model, a large operator sells services (like IPTV, Internet, IP phone, and wireless) to smaller operators. These other operators might be broadcasters, ISPs, content producers, telcos, cable companies — or whatever. They in turn resell these services under their own distinct brands to end-user consumers.

One branding element might be a different "skin" for set top box navigation — i.e., different colors, graphics, and information layout. Another might be more flexible business rules — e.g., longer "happy hours" or more liberal 2-for-1 movie rental promotions for just-released titles.

The key to making this model work is the ability to easily, rapidly, and cost-effectively tailor and blend services differently for different customer sets — *on a common platform* to achieve economies of scale. Those economies disappear if operators have to implement redundant platforms on which to base services that look or behave differently. For example, they would not want to implement redundant BSS/OSS connections (or possibly implement redundant BSS/OSSes). Nor would they want to install multiple IPTV systems just because its rules engine only works if you manage content assets a certain way (because rules and assets are tightly coupled). Nor would they want to install redundant IPTV back offices and gaming back offices (or a you-fill-in-the-blank back office) for the same reason.

The multi-business model is the ultimate test of whether your service delivery platform is truly universal. If your platform can support this model (even if you do not happen to deploy that model yourself) then you know it can very likely support the business model you have now or are likely to have in the future. Whatever your business model, you have the power to differentiate multiple operations throughout the value chain all the way from content to final application deployment.

Choice Rules

As the multi-business model illustrates, the power of your technology base ultimately rests with the ability to satisfy consumer choices at will. Just as consumers now want video content on-demand, in an IMS world they will want the next new hot application (whatever that may be) on-demand as well. Not having a service delivery platform that is universal will be like having your feet stuck in concrete.

A logical place to start is IPTV. Certain to be the centerpiece of any quad play offering, the right IPTV architecture gives consumers the rich multimedia experience they want now *and* gives you both capability and the experience at offering a very convergent product well before IMS actually arrives.

Knowing what to do next is always difficult, especially when technology moves so quickly and so many technology investments can be make-or-break. But with so many great IPTV applications here today, and so much hype about what IMS will mean tomorrow, standing still is not a choice.

For more information please contact your SeaChange representative or visit our website at www.schange.com.

SeaChange®