

SAAS - Software As A Service

SAAS

Software as a service (SaaS, typically pronounced 'Sass') is a model of software deployment where an application is hosted as a service provided to customers across the Internet. By eliminating the need to install and run the application on the customer's own computer, SaaS alleviates the customer's burden of software maintenance, ongoing operation, and support. Using SaaS also can reduce the up-front expense of software purchases, through less costly, on-demand pricing. From the software vendor's standpoint, SaaS has the attraction of providing stronger protection of its intellectual property and establishing an ongoing revenue stream. The SaaS software vendor may host the application on its own web server, or this function may be handled by a third-party application service provider (ASP). This way, end users may reduce their investment on server hardware too.

History

The concept of "software as a service" started to circulate prior to 1999 and was considered to be "gaining acceptance in the marketplace" in Bennett et al., 1999 paper on "Service Based Software"

Whilst the term "software as a service" was in common use, the camelback acronym "SaaS" was allegedly not coined until several years later in a white paper called "Strategic Backgrounder: Software as a Service" by the Software & Information Industry's eBusiness Division published in Feb. 2001, but written in fall of 2000 according to internal Association records.

Philosophy of SaaS

As a term, SaaS is generally associated with business software and is typically thought of as a low-cost way for businesses to obtain the same benefits of commercially licensed, internally operated software without the associated complexity and high initial cost. Consumer-oriented web-native software is generally known as Web 2.0 and not as SaaS. Many types of software are well suited to the SaaS model, where customers may have little interest or capability in software deployment, but do have substantial computing needs. Application areas such as Customer relationship management, video conferencing, human resources, IT service management, accounting and e-mail are just a few of the initial markets showing SaaS success. The distinction between SaaS and earlier applications delivered over the Internet is that SaaS solutions were developed specifically to leverage web technologies such as the browser, thereby making them web-native. The data design and architecture of SaaS applications are specifically built with a 'multi-tenant' backend, thus enabling multiple customers or users to access a shared data model. This further differentiates SaaS from client/server or 'ASP' (Application Service Provider) solutions in those SaaS providers are leveraging enormous economies of scale in the deployment, management, support and through the Software Development Lifecycle.

Key characteristics of software delivered by SaaS

The key characteristics of SaaS software, according to IDC, include:

- network-based access to, and management of, commercially available software
- activities that are managed from central locations rather than at each customer's site, enabling customers to access applications remotely via the Web
- application delivery that typically is closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering, and management characteristics
- Centralized feature updating, which obviates the need for downloadable patches and upgrades.

SaaS applications are generally priced on a per-user basis, sometimes with a relatively small minimum number of users and often with additional fees for extra bandwidth and storage. SaaS revenue streams to the vendor are therefore lower initially than traditional software license fees, but are also recurring, and therefore viewed as more predictable, much like maintenance fees for licensed software.

SaaS implementation

SAAS - Software As A Service

According to Microsoft, SaaS architectures generally can be classified as belonging to one of four "maturity levels," whose key attributes are configurability, multi-tenant efficiency, and scalability. Each level is distinguished from the previous one by the addition of one of those three attributes:

- **Level 1 - Ad-Hoc/Custom:** At the first level of maturity, each customer has its own customized version of the hosted application and runs its own instance of the application on the host's servers. Migration of a traditional non-networked or client-server application to this level of SaaS typically requires the least development effort and reduces operating costs by consolidating server hardware and administration.
- **Level 2 - Configurable:** The second maturity level provides greater program flexibility through configurable metadata, so that many customers can use separate instances of the same application code. This allows the vendor to meet the different needs of each customer through detailed configuration options, while simplifying maintenance and updating of a common code base.
- **Level 3 - Configurable, Multi-Tenant-Efficient:** The third maturity level adds multi-tenancy to the second level, so that a single program instance serves all customers. This approach enables more efficient use of server resources without any apparent difference to the end user, but ultimately is limited in its scalability.
- **Level 4 - Scalable, Configurable, Multi-Tenant-Efficient:** At the fourth and final SaaS maturity level, scalability is added through a multi-tier architecture supporting a load-balanced farm of identical application instances, running on a variable number of servers. The system's capacity can be increased or decreased to match demand by adding or removing servers, without the need for any further alteration of application software architecture.

Virtualization also may be used in SaaS architectures, either in addition to multi-tenancy, or in place of it. One of the principal benefits of virtualization is that it can increase the system's capacity without additional programming. On the other hand, a considerable amount of programming may be required to construct a more efficient, multi-tenant application. Combining multi-tenancy and virtualization provides still greater flexibility to tune the system for optimal performance. In addition to full operating system-level virtualization, other virtualization techniques applied to SaaS include application virtualization and virtual appliances.

Various types of software components and frameworks may be employed in the development of SaaS applications. These tools can reduce the time to market and cost of converting a traditional on-premise software product or building and deploying a new SaaS solution. Examples include components for subscription management, grid computing software, web application frameworks, and complete SaaS platform products.

SaaS adoption

Drivers for SaaS adoption

The traditional rationale for outsourcing of IT systems is that by applying economies of scale to the operation of applications, a service provider can offer better, cheaper, more reliable applications than companies can themselves. The use of SaaS-based applications has grown dramatically, as reported by many of the analyst firms that cover the sector. But it's only in recent years that SaaS has truly flourished. Several important changes to the way we work have made this rapid acceptance possible.

- **Everyone has a computer:** Most information workers have access to a computer and are familiar with conventions from mouse usage to web interfaces. As a result, the learning curve for new, external applications is lower and less hand-holding by internal IT is needed.
- **Computing itself is a commodity:** In the past, corporate mainframes were jealously guarded as strategic advantages. More recently, the applications were viewed as strategic. Today, people know it's the business processes and the data itself—customer records, workflows, and pricing information—that matters. Computing and application licenses are cost centers, and as such, they're suitable for cost reduction and outsourcing. The adoption of SaaS could also drive Internet-scale to become a commodity.
- **Insourcing IT systems require expensive overhead** including salaries, health care, liability and physical building space.

SAAS - Software As A Service

- Applications are standardized: With some notable, industry-specific exceptions, most people spend most of their time using standardized applications. An expense reporting page, an applicant screening tool, a spreadsheet, or an e-mail system are all sufficiently ubiquitous and well understood that most users can switch from one system to another easily. This is evident from the number of web-based calendaring, spreadsheet, and e-mail systems that have emerged in recent years.
- Parametric applications are usable: In older applications, the only way to change a workflow was to modify the code. But in more recent applications—particularly web-based ones—significantly new applications can be created from parameters and macros. This allows organizations to create many different kinds of business logic atop a common application platform. Many SaaS providers allow a wide range of customization within a basic set of functions.
- A specialized software provider can target global markets: A company that made software for human resource management at boutique hotels might once have had a hard time finding enough of a market to sell its applications. But a hosted application can instantly reach the entire market, making specialization within a vertical not only possible, but preferable. This in turn means that SaaS providers can often deliver products that meet their markets' needs more closely than traditional "shrinkwrap" vendors could.
- Web systems are reliable enough: Despite sporadic outages and slow-downs, most people are willing to use the public Internet, the Hypertext Transfer Protocol and the TCP/IP stack to deliver business functions to end users.
- Security is sufficiently well trusted and transparent: With the broad adoption of SSL organizations have a way of reaching their applications without the complexity and burden of end-user configurations or VPNs.
- Availability of enablement technology: According to IDC, organizations developing enablement technology that allow other vendors to quickly build SaaS applications will be important in driving adoption. Because of SaaS' relative infancy, many companies have either built enablement tools or platforms or are in the process of engineering enablement tools or platforms. A Saugatuck study shows that the industry will most likely converge to three or four enablers that will act as SaaS Integration Platforms (SIPs).
- Wide Area Network's bandwidth has grown drastically following Moore's Law (more than 100% increase each 24 months) and is about to reach slow local networks bandwidths. Added to network quality of service improvement this has driven people and companies to trustfully access remote locations and applications with low latencies and acceptable speeds.

Factors Limiting SaaS adoption

This section is missing citations or needs footnotes.

Using inline citations helps guard against copyright violations and factual inaccuracies. (May 2008)

SaaS is based upon the assumption that the services provided are commonplace and well defined, hence economies of scale and balancing of supply and demand becomes possible. This assumption holds true for those areas of IT that are ubiquitous, a cost of doing business and commodity-like. SaaS is therefore not suitable for innovative or highly specialized niche systems, though SaaS may be used to provide one or more components in such systems.

As with manufacturing, a lack of substitutability and second sourcing options with any commodity creates a strategic weakness for any customer in terms of security, competition and pricing. Various forms of this weakness, such as 'vendor lock-in', are often cited as a barrier to adoption of SaaS as the current industry lacks portability and interoperability between vendors. This means that to change from one vendor to another will take a considerable amount of effort and time. This situation is resolvable by the introduction of open sourced standards and the development of markets based upon such standards.

Whilst the severe lack of substitutability is unresolved, many vendors counter the concerns over potential security and operational risk with the argument that the professionals operating SaaS applications may have much better security and redundancy tools available to them.

SAAS - Software As A Service

Furthermore the concern that SaaS applications pose some difficulty for businesses that need extensive customization is countered with the claim that many vendors have made progress with both customization and publication of their programming interfaces. It should be noted that customization will reduce substitutability and given that SaaS covers commodity-like activities, the strategic benefit of customization is highly dubious.

In addition to this, the availability of open source applications, inexpensive hardware and low cost bandwidth combine to offer compelling economic reasons for businesses to operate their own software applications, particularly as open source solutions have become higher quality and easier to install.

SaaS Sales Channels

With products below the \$100 range and its focus on the mid market, direct selling can become an expensive undertaking. SaaS companies are seeking alternatives by selling through value-added resellers (VARs), Managed Service Providers (MSPs), Master Managed Service Providers (MMSPs) and similar alliance partners. But since SaaS is not only a different delivery mechanism but a different business model and different technology as well, selling through channels has its own challenges. A recent white paper published by the SIIA (Software & Information Industry Association) explains such differences to traditional software in more details

SaaS Monetization

The Anatomy of a Good SaaS Monetization Platform

SaaS providers should achieve their revenue acceleration, revenue assurance and improved operational efficiency goals by using a SaaS metering solution. The monetization platform should be built on a set of innovative software modules which meter usage of on-demand applications without the need for custom coding.

SaaS providers benefit because the monetization platform will:

- Ensure revenue stream by developing a deeper understanding of customers' needs and expectations. Hence, have a better customer focus.
- Recognize efficiencies and reduce costs by focusing on their core competency.
- By outsourcing the development of an operational platform they'll improve operational efficiencies and enjoy cost savings
- Expand market presence and increase revenue by creating service bundles for a variety of budgets

A good product will directly address the SaaS providers' pain points such as:

- insufficient service usage data
- distraction of R&D focus from core service development initiatives
- inability to support flexible pricing and billing models
- difficulty servicing customers
- higher costs for manual contract management
- inability to attract new customers to grow market share

The foundation of a good product offering is an Operations Service Manager (OSM) delivered as SaaS. The foundation provides the underlying security, scalability, data storage and visual rendering capabilities. The OSM aggregates customer, business service and service usage data for business processes. The Price Catalog, Usage Tracking, Contract Management and Billing Mediation are the business modules which run on the OSM to provide the necessary functionality.

Pricing Catalog: Publish Business Services

SaaS providers need to advertise services like an auto parts company does in a catalog or a restaurant does in its menu.

SAAS - Software As A Service

Their marketing and sales teams need the ability to bundle features, create specials, provide custom orders and clearly communicate this information to their customers. From purely a pricing perspective, providers should consider deconstructing their products into a list of features, associate a price to each feature and then re-construct the product into bundles of features or packages and define how the package is priced. Good monetization software provides an easy to use yet powerful Business Service Catalog that gives providers the flexibility they need to create and position bundled products.

- Benefit to Product Management – Product Managers can prioritize features for competitive reasons and price it to maximize revenue with actual usage data
- Benefit to Sales – Sales personnel can bundle features by usage tiers and provide pricing for a variety of budgets
- Benefit to R&D – Development groups can continue to develop the core business services and not worry about the sales operational infrastructure

Usage Tracking of SaaS: Understand Service Utilization

SaaS providers need to measure service usage for pricing consideration, billing purposes and improving revenue generating features. Providers have a big advantage when their customers use systems managed by them. However this benefit can only be realized if they effectively track customers' usage. SaaS Providers can make informed product enhancement decisions, empower the Sales team and ensure contract compliance using easy-to-understand, detailed service usage data. Knowing customer habits is vital for success.

A good monetization platform must have a Usage Tracking module that allows SaaS providers on heterogeneous platforms to collect service usage data and visualize it in a manner most useful to their business. Good technology allows for seamless integration with a provider's usage repository and aggregates the information on an easy to use dashboard. It gives providers the ability to view how customers are utilizing the business service – not page hits or similar high level web statistics, but a detailed view of how specific customers are using the most important features.

- Benefit to Sales – Sales teams can determine how prospects and customers are utilizing the service to help with up selling and renewals
- Benefit to Product Management – Product Managers can prioritize or accelerate development new features based on actual utilization data
- Benefit to Finance – Finance departments can gauge if customers are within usage agreement boundaries
- Benefit to Customer Service – Customer service teams can create support documentation for the most used features
- Benefit to your Customers – Customers can see how much of the service they are actually using
- Benefit to R&D – Development groups can now continue to develop the core business services and not worry about the usage monitoring infrastructure

Contract Management: Automate Contract Generation and Compliance

SaaS providers need to generate contracts for cataloged services and ensure compliance. SaaS is all about providing business value to customers based on their needs. To do this, SaaS providers need to provide a-la-carte pricing and bundle features into sellable packages. There are many benefits to usage based pricing; however, it also brings complexity to contract creation, tracking and enforcement. A good monetization platform delivers the ability to simplify the creation, metering and compliance of contracts and allows SaaS providers to reap the benefits of not having to deal with the contract complexities. A Contract Management module will allow SaaS providers to create and track unique customer contracts based on packages or custom pricing.

- Benefit to Sales – Sales personnel can quickly create usage contracts based on packages and customize as needed.
- Benefit to Finance – Finance can automate storage and tracking of custom usage-based agreements
- Benefit to R&D – Development groups can continue to focus on their core business services and not deal with the contract management infrastructure

SAAS - Software As A Service

Billing Mediation Simplify the Billing Process

SaaS providers need to generate invoices in accordance with contracts. Generating usage based invoices manually is time consuming and error prone. Therefore, good SaaS monetization software will have a Billing Mediation module that can automate this process and create invoices based on customer order and billing cycles. It then optionally sends invoice data to a payment processing system through an integration infrastructure to any major accounting system.

- Benefit to Finance – Finance and Accounting Departments can generate invoices based on actual usage and integrate with existing billing or ERP systems
- Benefit to R&D – Development groups can continue to focus on their core business services and not deal with the billing infrastructure

Pricing Strategy

Pricing strategy is a major component of any business, whether brick-and-mortar or bits-and-bytes. Generally, it's quite tricky to do correctly. Should pricing be value based or cost based? Should it focus on maximizing income per unit or volume sold? For all of the things the SaaS delivery model does for us, it does not ease pricing strategy pains.

So, how should a SaaS offering be priced? Clearly, this is something that should be considered on a vendor by vendor basis. However, there is a good strategy that focuses on boosting adoption rates, creation of income and healthy margins.

In perpetual license models, high prices generally posed significant barriers to on-boarding new customers, whether it was directly because of the expense or indirectly because of things like requiring approval from line of business managers, the C-level executive or purchasing. SaaS has significantly reduced this barrier but brought its own problems: one single sale of a perpetual license generally equated to significant revenue while one single SaaS customer does not. Given the aforementioned, many providers strive to rapidly boost adoption to reach a sustainable cash-flow position, and rightfully so. But selling at low prices and giving unrestricted access to everything leaves lots of money on the table. There is a solution!

The Strategy

A good SaaS pricing strategy provides the ability to achieve five objectives:

- Quantify usage patterns by capturing empirical data
- Mine the usage data and apply BI tools to extrapolate new "access bundles"
- Boost Adoption rates of new customers
- Control Cost for acquiring new customers and up-selling the existing base
- Generate Healthy Margins and Profit

Adoption is generally boosted by lowering the barrier (price) to acquire your SaaS offerings functionality. The optimal base price plan(s) is one that includes a low price and access to all features of your product, but, has contractual limitations on the number of times the client may access specific features. Any additional access can be done, without interruption to the user, but is logged as a "for-charge" use overage. The cost for the overage charges should be slightly above those contracted for in the base plan. This is very similar to your cell phone calling plan or your office photo-copier plan. Such plans reduce adoption resistance, optimize your ability to bring on new customers and maximize revenues for the services delivered. Therefore, usage metering is key to your success!

Generally, companies are not interested in selling at a loss. Obviously, you shouldn't have too many operating surprises and, given a good amount of time, your cost estimates should be quite accurate. This is where using the right monitoring software matters. By knowing EXACTLY what features your customers are using (when and how) delivers predictability and cost control.

SAAS - Software As A Service

Tracking usage patterns for each client allows up-selling of "access bundles" that allow clients to provision their fit usage patterns precisely and generate the greatest profit for your SaaS offering. The idea (see item 1 above) is to use a low cost oriented pricing to boost adoption then up-sell those clients into a cost justifiable "access bundle". You'll have a healthy market for up-selling your existing customers. And such up-selling poses significantly lower resistance to spending more money because it is specifically based on "real" usage data that you've collected; giving you good justification to charge a value oriented price.

Summary

The strategy is really very simple:

- Attract new customers to your base product offer and then up-sell these clients into high margin, value priced "access bundles" once they're on-board.
- Focus on getting as, but it seems to be a good starting point. The goal is to exploit the ease at which people can sign-up for SaaS offerings and get as many tenants into your easier to harvest tenant environment as possible. For more information on this subject Google "SaaS monetizing", "SaaS Metering", "SaaS Usage", "SaaS Pricing Strategies".

A SaaS Platform is a computer program or collection of more than one computer program that acts as a host to applications that reside on it. The platform manages underlying computer hardware and software resources and uses those resources to endow its hosted applications with multi-tenant, on-demand capabilities that are found in Software as service applications. Generally, hosted applications are written to target the platform and support a single user. The platform absorbs the responsibility of distributing the application as a service to multiple users over the Internet. The SaaS Platform can be considered a layer of abstraction above the traditional application server, creating a computing platform that parallels the value offered by the traditional operating system, only in a web-centric fashion. The SaaS platform is rooted in the need to reduce the time and difficulty associated with developing highly available, enterprise grade business applications that are to be delivered on-demand.

Online or Offline platforms

SaaS can be delivered either on an Online or an Offline platform. Some applications have the ability to work on both platforms.

Responsibilities

Aside from enhancing a hosted application with one-to-many capabilities, SaaS platforms generally take on a larger set of responsibilities. The list below highlights some of these.

Tenancy Partitioning

One of the major responsibilities of any SaaS platform is offering tenancy partitioning services. Applications written in a single tenant fashion generally have little or no concept of serving multiple customers. For example, the application database schema may be geared for catering to a single tenant (customer), preventing multiple customers from storing data in the same database. Furthermore, execution may not be partitioned so situations may arise where inadvertent state sharing occurs between tenants that otherwise should have been isolated from one another. SaaS platforms are responsible for providing multi-tenancy aspects when needed.

Scaling

SaaS applications purposefully aggregate demand for all customers and users to one physical or virtual location. Being able to support this aggregation is therefore a requirement of any SaaS application. To achieve this sort of scaling, the application must be designed in a fashion conducive to scaling as well as having the support required of any auxiliary

SAAS - Software As A Service

pieces. SaaS platforms attempt to remove the work associated with scaling by commoditizing the scaling portion

Monitoring & Metering

A SaaS application is required to meter and monitor usage from both data and execution standpoints. This is generally closely tied to both scaling and monetization models. Much of this is extracted into a platform layer, where the platform becomes responsible for metering tenant and user usage, as well as monitoring for system events.

SaaS providers achieve their revenue acceleration, revenue assurance and improved operational efficiency goals by using a SaaS metering solution. The monetization platform should be built a set of integrated software modules which meter usage of on-demand applications without the need for custom coding.

SaaS Monitoring & Metering Benefits

SaaS providers benefit because the monetization platform will:

- Ensure revenue stream by developing a deeper understanding of customers' needs and expectations
- Recognize efficiencies and reduce costs by focusing on their core competency
- Expand market presence and increase revenue by creating service bundles for a variety of budgets

A good platform will directly address the SaaS providers' pain points such as insufficient service usage data, distraction of R&D focus from core service development initiatives, inability to support flexible pricing and billing models, difficulty servicing customers, higher costs for manual contract management, and therefore, inability to attract new customers to grow market share.

The foundation of a good platform offering is an Operations Service Manager (OSM) delivered as SaaS. The foundation provides the underlying security, scalability, data storage and visual rendering capabilities. The OSM aggregates customer, business service and service usage data for business processes. The Price Catalog, Usage Tracking, Contract Management and Billing Mediation are the business modules which run on the OSM to provide the necessary functionality. Pricing Catalog: Publish Business Services

SaaS providers need to advertise services like an auto parts company does in a catalog or a restaurant does in its menu. Their marketing and sales teams need the ability to bundle features, create specials, provide custom orders and clearly communicate this information to their customers. From purely a pricing prospective, providers should consider deconstructing their products into a list of features, associate a price to each feature and then re-construct the product into bundles of features or packages and define how the package is priced. Good monetization software provides an easy to use yet powerful Business Service Catalog that gives providers the flexibility they need to create and position bundled products.

Usage Tracking: Understand Service Utilization

SaaS providers need to measure service usage for pricing consideration, billing purposes and improving revenue generating features. Providers have a big advantage when their customers use systems managed by them. However this benefit can only be realized if they effectively track customers' usage. SaaS Providers can make informed product enhancement decisions, empower the Sales team and ensure contract compliance using easy-to-understand, detailed service usage data. Knowing customer habits is vital for success.

A good monetization platform must have a Usage Tracking module that allows SaaS providers on heterogeneous platforms to collect service usage data and visualize it in a manner most useful to their business. Good technology allows for seamless integration with a provider's usage repository and aggregates the information on an easy to use dashboard. It gives providers the ability to view how customers are utilizing the business service – not page hits or similar high level web statistics, but a detailed view of how specific customers are using the most important features.

SAAS - Software As A Service

Contract Management: Automate Contract Generation and Compliance

SaaS providers need to generate contracts for cataloged services and ensure compliance. SaaS is all about providing business value to customers based on their needs. To do this, SaaS providers need to provide a-la-carte pricing and bundle features into sellable packages. There are many benefits to usage based pricing; however, it also brings complexity to contract creation, tracking and enforcement. A good monetization platform delivers the ability to simplify the creation, metering and compliance of contracts and allows SaaS providers to reap the benefits of not having to deal with the contract complexities. A Contract Management module will allow SaaS providers to create and track unique customer contracts based on packages or custom pricing.

Billing Mediation Simplify the Billing Process

SaaS providers need to generate invoices in accordance with contracts. Generating usage based invoices manually is time consuming and error prone. Therefore, good SaaS monetization software will have a Billing Mediation module that can automate this process and create invoices based on customer order and billing cycles. It then optionally sends invoice data to a payment processing system through an integration infrastructure to any major accounting system.

Distributed Services

Traditional, on-premise software relies on a variety of services to accomplish the task of executing business logic. These on-premise services, such as logging or job scheduling, tend to become overly complex if used in a distributed system that generally backs most SaaS applications. SaaS platforms provide distributed equivalents of many such services, giving similar or even augmented capability when comparing to on-premise counterparts.

SaaS platform companies are currently introducing themselves to the market. Generally, their platforms can be categorized based on focus: either vertically aligned, horizontally aligned, product-centric, or general purpose. Although each is categorically different, the platforms generally adhere to most of the basic principals outlined above.

Lock-in

One topic that frequently surfaces within SaaS and SaaS platform circles is the topic of lock-in. Much like other technologies, SaaS platforms offer varying degrees of openness. On the proprietary side, an example is the upcoming Salesforce.com's AppExchange and it's underlying Apex platform, which requires the use of a proprietary programming language and technology stack. Generally, this is considered a lock-in approach to the platform concept. Lock-in through new programming languages and frameworks tend to increase adoption risk and require that any existing technology (such as open source components) be ignored or integrated in through adapter mechanisms if possible. Arguments supporting lock-in style platforms tend to originate from the claim that a new language and framework is needed to support SaaS applications.

Conversely, SaaS platform newcomers such as Apprenda and SaaSWizard, have announced intentions to introduce platforms to the market that present little to no technological lock-in. The goal is to allow the use of existing technologies to provide implementation logic, while attaching SaaS functionality to the software through the platform. Criticisms tend to focus on the fact that providing SaaS functionality in an easy-to-implement fashion is difficult to do using existing technologies, and could prove burdensome for providers.