



CLOUD COMPUTING

Unit-4

Introduction to SaaS, Web Services, Web 2.0



AVINAV PATHAK

Assistant Professor

***Shobhit Institute of Engineering & Technology
(Deemed-to-be-University), Meerut, India***

Introduction to SaaS



Software-as-a-Service (SaaS) model allows to provide software application as a service to the end users. It refers to a software that is deployed on a host service and is accessible via Internet. There are several SaaS applications listed below:

- 1. Billing and invoicing system***
- 2. Customer Relationship Management (CRM) applications***
- 3. Help desk applications***
- 4. Human Resource (HR) solutions***
- 5. Some of the SaaS applications are not customizable such as Microsoft Office Suite. But SaaS provides us Application Programming Interface (API), which allows the developer to develop a customized application.***

Introduction to SaaS



Characteristics

- 1. Here are the characteristics of SaaS service model:***
- 2. SaaS makes the software available over the Internet.***
- 3. The software applications are maintained by the vendor.***
- 4. The license to the software may be subscription based or usage based. And it is billed on recurring basis.***
- 5. SaaS applications are cost-effective since they do not require any maintenance at end user side.***
- 6. They are available on demand.***
- 7. They can be scaled up or down on demand.***
- 8. They are automatically upgraded and updated.***

Introduction to SaaS



Benefits

Using SaaS has proved to be beneficial in terms of scalability, efficiency and performance. Some of the benefits are listed below:

1.Modest software tools

2.Efficient use of software licenses

3.Centralized management and data

4.Platform responsibilities managed by provider

5.Multitenant solutions

SaaS Issues



Modest software tools

- **The SaaS application deployment requires a little or no client side software installation, which results in the following benefits:**
- **No requirement for complex software packages at client side**
- **Little or no risk of configuration at client side**
- **Low distribution cost**

Efficient use of software licenses

The customer can have single license for multiple computers running at different locations which reduces the licensing cost. Also, there is no requirement for license servers because the software runs in the provider's infrastructure.

Centralized management and data

The cloud provider stores data centrally. However, the cloud providers may store data in a decentralized manner for the sake of redundancy and reliability.

SaaS Issues



Platform responsibilities managed by providers

All platform responsibilities such as backups, system maintenance, security, hardware refresh, power management, etc. are performed by the cloud provider. The customer does not need to bother about them.

Multitenant solutions

Multitenant solutions allow multiple users to share single instance of different resources in virtual isolation. Customers can customize their application without affecting the core functionality.

SaaS and SOA



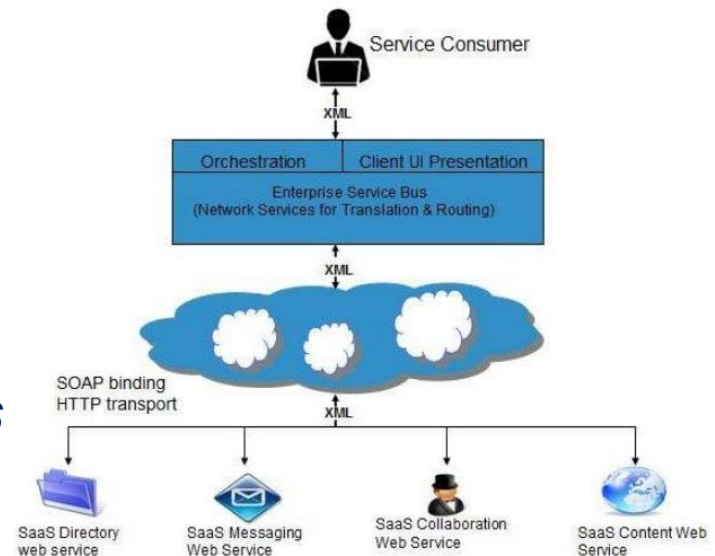
Open SaaS and SOA

Open SaaS uses those SaaS applications, which are developed using open source programming language. These SaaS applications can run on any open source operating system and database.

Open SaaS has several benefits

listed below:

- **No License Required**
- **Low Deployment Cost**
- **Less Vendor Lock-in**
- **More portable applications**
- **More Robust Solution**
- **The following diagram shows the SaaS implementation based on SOA:**





Web Services

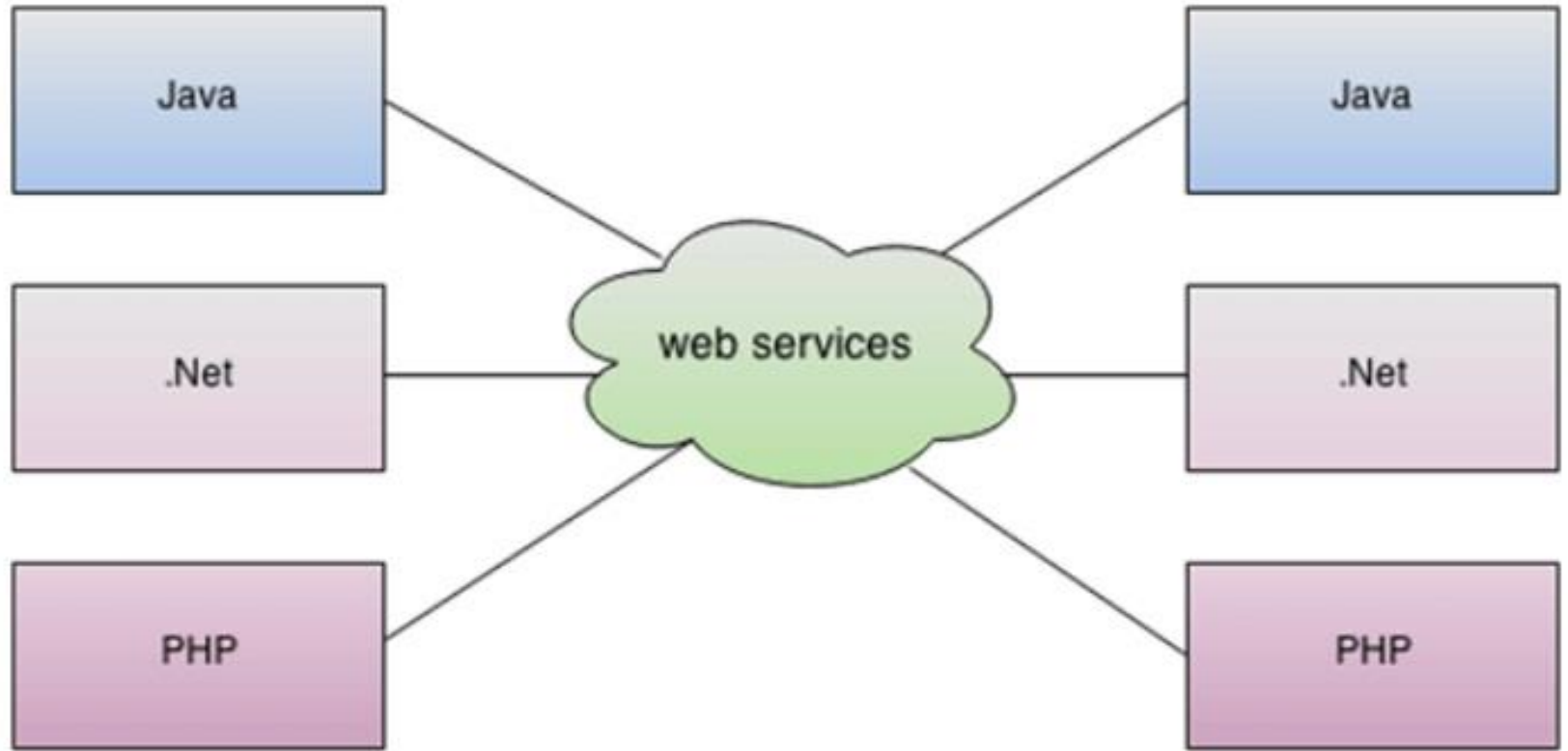
What is Web Service?

A Web Service is can be defined by following ways:

- 1. It is a client-server application or application component for communication.**
- 2. The method of communication between two devices over the network.**
- 3. It is a software system for the interoperable machine to machine communication.**
- 4. It is a collection of standards or protocols for exchanging information between two devices or application.**

Let's understand it by the figure given below: As you can see in the figure, Java, .net, and PHP applications can communicate with other applications through web service over the network. For example, the Java application can interact with Java, .Net, and PHP applications. So web service is a language independent way of communication.

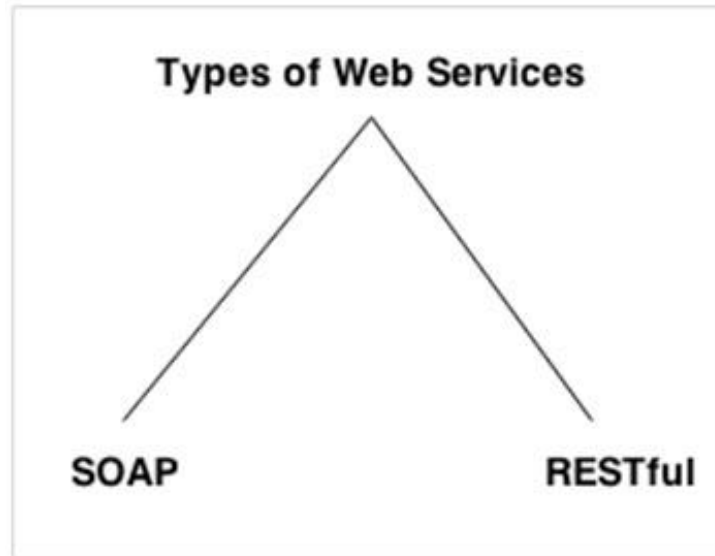
Web Services



Types of Web Services

There are mainly two types of web services.

- 1. SOAP web services.***
- 2. RESTful web services.***



Types of Web Services



Web Service Features

XML-Based

Web services use XML at data description and data transportation layers. Using XML exclude any networking, operating system, or platform binding. Web services-based operation is extremely interoperable at their core level.

Loosely Coupled

A client of a web service is not fixed to the web service directly. The web service interface can support innovation over time without negotiating the client's ability to communicate with the service. A tightly coupled system means that the client and server logic are closely tied to one another, indicating that if one interface changes, then another must be updated. Accepting a loosely coupled architecture tends to make software systems more manageable and allows more straightforward integration between various systems.

Types of Web Services



Web Service Features

Coarse-Grained

Object-oriented technologies such as Java expose their functions through individual methods. A specific process is too fine an operation to provide any suitable capability at a corporate level. Building a Java program from scratch needed the creation of various fine-grained functions that are then collected into a coarse-grained role that is consumed by either a client or another service.

Businesses and the interfaces that they prove should be coarse-grained. Web services technology implement a natural method of defining coarse-grained services that approach the right amount of business logic.



Types of Web Services

Ability to be Synchronous or Asynchronous

Synchronicity specifies the binding of the client to the execution of the function. In synchronous invocations, the client blocks and delays in completing its service before continuing. Asynchronous operations grant a client to invoke a task and then execute other functions.

Asynchronous clients fetch their result at a later point in time, while synchronous clients receive their effect when the service has completed. Asynchronous capability is an essential method in enabling loosely coupled systems.

Supports Remote Procedure Calls (RPCs)

Web services allow consumers to invoke procedures, functions, and methods on remote objects using an XML-based protocol. Remote systems expose input and output framework that a web service must support.



Types of Web Services

Component development through Enterprise JavaBeans (EJBs) and .NET Components has more become a part of architectures and enterprise deployments over a previous couple of years. Both technologies are assigned and accessible through a variety of RPC mechanisms.

A web function supports RPC by providing services of its own, equivalent to those of a traditional role, or by translating incoming invocations into an invocation of an EJB or a .NET component.

Supports Document Exchange

One of the essential *benefits* of XML is its generic way of representing not only data but also complex documents. These documents can be as simple as describing a current address, or they can be as involved as defining an entire book or Request for Quotation (RFQ). Web services support the transparent transfer of documents to facilitate business integration.

Web Service Components



There are three major web service components.

- 1. SOAP**
- 2. WSDL**
- 3. UDDI**

SOAP

- **SOAP is an acronym for Simple Object Access Protocol.**
- **SOAP is a XML-based protocol for accessing web services.**
- **SOAP is a W3C recommendation for communication between applications.**

SOAP is XML based, so it is platform independent and language independent. In other words, it can be used with Java, .Net or PHP language on any platform.

WSDL

- **WSDL is an acronym for Web Services Description Language.**
- **WSDL is a xml document containing information about web services such as method name, method parameter and how to access it.**
- **WSDL is a part of UDDI. It acts as a interface between web service applications.**
- **WSDL is pronounced as wiz-dull.**

UDDI

- **UDDI is an acronym for Universal Description, Discovery and Integration.**
- **UDDI is a XML based framework for describing, discovering and integrating web services.**
- **UDDI is a directory of web service interfaces described by WSDL, containing information about web services.**



SOAP Web Services

SOAP stands for Simple Object Access Protocol. It is a XML-based protocol for accessing web services.

SOAP is a W3C recommendation for communication between two applications.

SOAP is XML based protocol. It is platform independent and language independent. By using SOAP, you will be able to interact with other programming language applications.

Advantages of Soap Web Services

- **WS Security: SOAP defines its own security known as WS Security.**
- **Language and Platform independent: SOAP web services can be written in any programming language and executed in any platform.**

Disadvantages of Soap Web Services

- **Slow: SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.**
- **WSDL dependent: SOAP uses WSDL and doesn't have any other mechanism to discover the service.**



RESTful Web Services

RESTful Web Services

REST stands for REpresentational State Transfer.

REST is an architectural style not a protocol.

Advantages of RESTful Web Services

- ***Fast: RESTful Web Services are fast because there is no strict specification like SOAP. It consumes less bandwidth and resource.***
- ***Language and Platform independent: RESTful web services can be written in any programming language and executed in any platform.***
- ***Can use SOAP: RESTful web services can use SOAP web services as the implementation.***
- ***Permits different data format: RESTful web service permits different data format such as Plain Text, HTML, XML and JSON.***

Web 2.0



Web 2.0 (also known as Participative (or Participatory) and Social Web refers to websites that emphasize user-generated content, ease of use, participatory culture and interoperability (i.e., compatible with other products, systems, and devices) for end users.

The term was invented by Darcy DiNucci in 1999 and later popularized by Tim O'Reilly and Dale Dougherty at the O'Reilly Media Web 2.0 Conference in late 2004. The Web 2.0 framework specifies only the design and use of websites and does not place any technical demands or specifications on designers. The transition was gradual and, therefore, no precise date for when this change happened has been given.



Web 2.0

A Web 2.0 website allows users to interact and collaborate with each other through social media dialogue as creators of user-generated content in a virtual community. This contrasts the first generation of Web 1.0-era websites where people were limited to viewing content in a passive manner. Examples of Web 2.0 features include social networking sites or social media sites (e.g., Facebook), blogs, wikis, folksonomies ("tagging" keywords on websites and links), video sharing sites (e.g., YouTube), hosted services, Web applications ("apps"), collaborative consumption platforms, and mashup applications.

Whether Web 2.0 is substantially different from prior Web technologies has been challenged by World Wide Web inventor Tim Berners-Lee, who describes the term as jargon. His original vision of the Web was "a collaborative medium, a place where we [could] all meet and read and write." On the other hand, the term Semantic Web (sometimes referred to as Web 3.0)[10] was coined by Berners-Lee to refer to a web of content where the meaning can be processed by machines.

Web 2.0 Characteristics



The key features of Web 2.0 include:

- 1. Folksonomy – free classification of information; allows users to collectively classify and find information (e.g. "tagging" of websites, images, videos or links)**
- 2. Rich user experience – dynamic content that is responsive to user input (e.g., a user can "click" on an image to enlarge it or find out more information)**
- 3. User participation – information flows two ways between the site owner and site users by means of evaluation, review, and online commenting. Site users also typically create user-generated content for others to see (e.g., Wikipedia, an online encyclopedia that anyone can write articles for or edit)**
- 4. Software as a service (SaaS) – Web 2.0 sites developed APIs to allow automated usage, such as by a Web "app" (software application) or a mashup**
- 5. Mass participation – near-universal web access leads to differentiation of concerns, from the traditional Internet user base (who tended to be hackers and computer hobbyists) to a wider variety of users**

Web 2.0 Concepts



Web 2.0 can be described in three parts:

➤ **Rich Internet application (RIA)** — defines the experience brought from desktop to browser, whether it is "rich" from a graphical point of view or a usability/interactivity or features point of view.

➤ **Web-oriented architecture (WOA)** — defines how Web 2.0 applications expose their functionality so that other applications can leverage and integrate the functionality providing a set of much richer applications. Examples are feeds, RSS feeds, web services, mashups.

➤ **Social Web** — defines how Web 2.0 websites tend to interact much more with the end user and make the end user an integral part of the website, either by adding his or her profile, adding comments on content, uploading new content, or adding user-generated content (e.g., personal digital photos).

As such, Web 2.0 draws together the capabilities of client- and server-side software, content syndication and the use of network protocols. Standards-oriented Web browsers may use plug-ins and software extensions to handle the content and user interactions. Web 2.0 sites provide users with information storage, creation, and dissemination capabilities that were not possible in the environment known as "Web 1.0".



Web 2.0 Concepts

Web 2.0 sites include the following features and techniques, referred to as the acronym SLATES by Andrew McAfee:

➤ **Search**

Finding information through keyword search.

Links to other websites

Connects information sources together using the model of the Web.

➤ **Authoring**

The ability to create and update content leads to the collaborative work of many authors. Wiki users may extend, undo, redo and edit each other's work. Comment systems allow readers to contribute their viewpoints.

➤ **Tags**

Categorization of content by users adding "tags" — short, usually one-word or two-word descriptions — to facilitate searching. For example, a user can tag a metal song as "death metal". Collections of tags created by many users within a single system may be referred to as "folksonomies" (i.e., folk taxonomies).

Web 2.0 Concepts



➤ Extensions

Software that makes the Web an application platform as well as a document server. Examples include Adobe Reader, Adobe Flash, Microsoft Silverlight, ActiveX, Oracle Java, QuickTime, and Windows Media.

➤ Signals

The use of syndication technology, such as RSS feeds to notify users of content changes.

While SLATES forms the basic framework of Enterprise 2.0, it does not contradict all of the higher level Web 2.0 design patterns and business models. It includes discussions of self-service IT, the long tail of enterprise IT demand, and many other consequences of the Web 2.0 era in enterprise uses.

Documents of Web 2.0 can be assessed by measures related to such quality dimension as accessibility, completeness, credibility, involvement, objectivity, readability, relevance, reputation, style, timeliness, uniqueness and usefulness



Web 2.0 Usage

A third important part of Web 2.0 is the social web. The social Web consists of a number of online tools and platforms where people share their perspectives, opinions, thoughts and experiences. Web 2.0 applications tend to interact much more with the end user. As such, the end user is not only a user of the application but also a participant by:

- 1. Podcasting***
- 2. Blogging***
- 3. Tagging***
- 4. Curating with RSS***
- 5. Social bookmarking***
- 6. Social networking***
- 7. Social media***
- 8. Wikis***
- 9. Web content voting: Review site or Rating site***



Web OS, Case Studies

Web OS



webOS, also known as LG webOS and previously known as Open webOS, HP webOS and Palm webOS, is a Linux kernel-based multitasking operating system for smart devices such as smart TVs that has also been used as a mobile operating system. Initially developed by Palm, Inc. (which was acquired by Hewlett-Packard), HP made the platform open source, at which point it became Open webOS. The operating system was later sold to LG Electronics. In January 2014, Qualcomm announced that it had acquired technology patents from HP, which included all the webOS and Palm patents.

Various versions of webOS have been featured on several devices since launching in 2009, including Pre, Pixi, and Veer smartphones, TouchPad tablet, LG's smart TVs since 2014, LG's smart refrigerators and smart projectors since 2017.

Features of HP Palm Web



OS

Multitasking interface

Navigation uses multi-touch gestures on the touchscreen. The interface uses "cards" to manage multitasking and represent apps. The user switches between running apps with a flick from left and right on the screen. Apps are closed by flicking a "card" up—and "off"—the screen. The app "cards" can be rearranged for organization. webOS 2.0 introduced 'stacks', where related cards could be "stacked" together.

Synergy

Palm referred to integration of information from many sources as "Synergy." Users can sign into multiple email accounts from different providers and integrate all of these sources into a single list. Similar capabilities pull together calendars and also instant messages and SMS text messages from multiple sources.

Over-the-air updates

The OS can be updated without docking to a PC, instead receiving OS updates over the carrier connection.

Features of HP Palm Web



OS On phones, when a notification comes in, it slides in from the bottom of the screen. Due to the resizable nature of the Mojo and Enyo application frameworks, the app usually resizes itself to allow unhindered use while the notification is displayed. After the notification slides away, it usually remains as an icon. The user can then tap on the icons to expand them. Notifications can then be dismissed (sliding off the screen), acted upon (tapping), or left alone.

Sync

By default, data sync uses a cloud-based approach rather than using a desktop sync client. The first version of webOS shipped with the ability to sync with Apple's iTunes software by masquerading as an Apple device, but this feature was disabled by subsequent iTunes software updates.

Third-party applications

On HP webOS, officially vetted third-party apps are accessible to be installed on the device from the HP App Catalog.

As HP webOS replaced Palm OS, Palm commissioned MotionApps to code and develop an emulator called Classic, to enable backward compatibility to Palm OS apps. This operates with webOS version 1.0. Palm OS emulation was discontinued in WebOS version 2.0. MotionApps disengaged from Classic in 2010, citing HP Palm as "disruptive."

Case Studies (SaaS)



Avantifix teams up with i-Virtuals to move to the cloud

Avantifix, a fixing solutions specialist has embraced modern technologies to help make a good business even better...

One of the problems facing any small firm is how much to invest in IT. After all, you've got a turnover measured in thousands rather than millions, even the purchase of a server and some PCs may seem like a step too far.

Mike McElhatton, the managing director of Avantifix, a distributor for fixing tools for the construction industry , was in that very position. Indeed, McElhatton, while appreciating the need for computers, is of the generation that is as happy with pen and paper. "I have very little interest in IT but it's part and parcel of the business," he says.

Case Studies (SaaS)



Avantifix teams up with i-Virtuals to move to the cloud

He didn't know how to progress with the IT system. "We had what I'd call a DIY system. We couldn't afford to have someone dedicated to IT but we could run what we had. For example, would I be able to build a barbecue out of bricks, yes. Would I build a house, no? It would take 10 years," he says.

Back then

Avantifix had a stock management system called Merlin— a DOS one no less, hosted on Windows 2003 server, which was clearly coming to the end of its natural life. The company did have a website but it didn't enable its customers to make purchases on account – and for a firm that made most of its sales to account customers, billing them later, that was a serious omission – one that was losing them business to rival firms.

McElhatton realised the system had to be upgraded but didn't really know how to proceed. The replacement of the company's financial controller turned out to be the key to the process as that led to the introduction of a new service provider, i-Virtuals.

Case Studies (SaaS)



Ashoka Reddy, i-Virtuals managing director quickly ascertained that Avantifix would be a perfect candidate for a cloud-based system: the company had little or no internal IT expertise and was urgent need of a more flexible infrastructure.

However, McElhatton was also concerned that the company was not being pushed down a path of buying more than it actually needed. “It’s like the old-style Nokia phone. That was perfect for just making phone calls: it just worked. A computer system doesn’t need to have any more that we needs doing.”

Making way for change

The internal system has now been moved to an i-Virtuals cloud, with the on-premise server being used as backup to provide added resilience. That in itself is a new departure for the company. In the past, they didn’t get backed up but, drawing on one of the advantages of the cloud, everything is now backed up automatically.

Case Studies (SaaS)



The new system allows Avantifix to run a network of 10 PCs interconnected, providing access to the Merlin stock control system and Sage accounting. The company also uses the system to connect the employees' iPads.

The new system also means that it is easier for Avantifix to stay on top of software licensing, an area that's often tricky for small businesses.

One step that Avantifix is still to deal with is the issue of its website. While there's an acceptance that it needs upgrading, there is the thorny problem of bringing together information from a variety of different sources, all of which have their own individual formats.

Reddy says that upgrading the website would mean developing a completely integrated system that's all linked to the same database, something that would be very time-consuming to implement.

As McElhatton explains this would be something that only he and his team could do as it's a task that requires human involvement.

Case Studies (SaaS)



Motorola improves compensation accuracy

Comms company works with Xactly SaaS to improve sales rep compensation scheme

If there's one thing that sales executives around the world are agreed on, it's on the importance of pay and commission. It's not always the easiest of processes to deal with - especially when there's an international sales force with a multitude of currencies. Motorola Solutions turned to the cloud in an attempt to simplify the process of dealing with a complex web of financial recompense.

As with many companies, the move to cloud arose after a company reorganisation. January 2011, Motorola Solutions spun-off the consumer focused side of its business to form Motorola Mobility enabling Motorola Solutions to focus on its core business as a provider of mission-and business-critical communication products and services. The company is one of the world's leading suppliers of rugged devices that deliver real-time information to both public safety operators (police, fire and ambulance) and businesses in markets as diverse as retail, shipping and hospitality. And it sells into more than 100 countries, employing 1700 sales personnel.

Challenge

It is this spread that provides Motorola Solutions with a major challenge. Lisa Bentley-Smith, sales incentive manager for EMEA, manages the compensation for 400 sales reps in what is an extremely diverse and complex region, having to deal with different currencies, languages, cultures and local laws.

Case Studies (SaaS)



Motorola improves compensation accuracy

When Bentley-Smith joined the sales management team in 2011, compensation was entirely manual and spreadsheet based. Not only was the process time-consuming for the admin staff, which had to deal with multiple data sources and a large number of queries, but also it was also frustrating for the sales team. With limited visibility over how they were being paid, personnel were often spending too much time checking whether they were being paid correctly.

“Walking down the halls, compensation was often the first thing to be mentioned by the general manager, the vice presidents and the directors,” says Bentley-Smith. “We wanted to end this – for compensation to become a non-issue.”

With managers spending increasing amounts of time on this activity it was clear that change was required. The EMEA region needed a new system that could fully automate the payment process and cope with all the complexities of the region in a simple, cost-effective manner.

“We decided to speak to our sales team to find out what they wanted from the new system,” says Bentley-Smith. “It turned out it was basic stuff. They wanted to know what they were going to be paid when and, if necessary, be able to raise queries quickly and efficiently.”