IT@Intel White Paper

Intel Information Technology
Cloud Computing
October 2009

Architecting Software as a Service for the Enterprise

The reference architecture provides a proven template solution that Intel SaaS project teams can immediately apply to specific implementation projects.

Catherine Spence
Enterprise Architect, Intel IT

Jason Devoys
Enterprise Architect, Intel IT

Sudip Chahal
Principal Engineer, Intel IT

Executive Overview

As part of our cloud computing strategy, Intel IT has been opportunistically taking advantage of external offerings of software as a service (SaaS) applications. To prepare for broader SaaS adoption, we designed a SaaS architecture that will enable us to shift to a more strategic view and facilitate faster, more standardized implementations.

To create our architecture, we analyzed SaaS industry trends and scanned existing Intel SaaS implementations to gather best-known methods and architectural techniques. We then extended existing enterprise application frameworks and architecture to create the elements that comprise the SaaS architecture, including:

- A use-case model based on a typical scenario that requires back-end data exchange between Intel and the SaaS provider.
- A conceptual architecture that provides a long-term view of all the components required in a complete SaaS implementation.
- A reference architecture providing a proven template solution that Intel SaaS project teams can immediately apply to specific implementation projects.

The SaaS architecture promotes standardization and best practices. It defines the components and capabilities required for deployment and a vocabulary for consistent communication with SaaS providers. Our goal is to facilitate a shift from organic growth to prescriptive deployment of SaaS applications at Intel, with greater consistency among implementations and reduced implementation effort.

Contents

IT@INTEL

IT@Intel is a resource that enables IT professionals, managers, and executives to engage with peers in the Intel IT organization—and with thousands of other industry IT leaders—so you can gain insights into the tools, methods, strategies, and best practices that are proving most successful in addressing today's tough IT challenges. Visit us today at www.intel.com/IT or contact your local Intel representative if you'd like to learn more.

BACKGROUND

Cloud computing is an important trend that includes several categories of service, all offered on demand over the Internet in a pay-as-you-go model. Software as a service (SaaS) is one of these categories; others include platform as a service (PaaS) and infrastructure as a service (IaaS). See the sidebar "A Cloud Computing Taxonomy" on page 3 for more details.

Intel IT has defined an overall cloud computing strategy based on growing the cloud from the inside out. We are developing a private internal cloud that will eventually extend to, and support interoperability with, the Internet or external cloud. Over time, this strategy will allow Intel to dynamically transfer workloads in and out of the enterprise, taking into account considerations such as cost, security, and compliance.

As we grow our internal cloud and determine how best to take advantage of the external cloud, we have also been opportunistically taking advantage of external SaaS offerings that deliver value such as increased agility and cost savings.

SaaS is the most mature category of cloud service, since it evolved from the application-service-provider model of software hosting. With SaaS, software applications are rented from a provider as opposed to purchased for enterprise installation and deployment.

Intel has successfully used a variety of SaaS applications. These typically have been specialized point solutions: Expense reporting was an early example, and adoption has grown as we have continued to identify specialized solutions that make sense to outsource. We've experienced accelerating growth of SaaS in the last few years. Based on the success of these solutions, we expect growth to continue.

The opportunistic use of SaaS has yielded benefits such as cost savings, improved agility, and faster time-to-market, as well as increased flexibility in scaling to support more users as necessary. It has also provided a venue for experimenting with new capabilities.

As our use of SaaS increases, we must be able to scale the environment, be positioned to take advantage of standardization, and provide guidance to suppliers about how to integrate with enterprise applications. In addition, future solutions will likely require more frequent data exchange between Intel and our providers, with lower tolerance for failure. We also look to improve security and manageability, which currently require a great deal of work, due diligence, and carefully calculated risk.

These requirements led us to develop a SaaS architecture that will help us shift to a more strategic view of SaaS and enable faster, more standardized implementations. Using the SaaS architecture, we can proactively assess applications in our portfolio, exploring opportunities to simplify our environment, gain faster access to new features, and reduce cost.

SAAS ARCHITECTURE

Our goal was to provide a proven template solution that comprehends the unique requirements of SaaS, defines the components and capabilities required for deployment, and promotes consistent communication with external solution providers.

We began by conducting an environmental scan of industry trends and existing Intel SaaS solution architecture. We then developed a series of architectural elements. These included a use-case model and a conceptual architecture that represent a long-term vision of the key capabilities required in a complete SaaS offering. Because not all of these capabilities are available today, we also developed a near-term reference architecture based on existing enterprise application frameworks and architecture. The reference architecture provides a proven template solution that Intel SaaS project teams can immediately apply to specific implementation projects.

Environmental Scan

The environmental scan incorporated two activities. First, we examined industry trends. Second, we surveyed representative Intel SaaS deployments to gather architectural techniques and best-known methods.

INDUSTRY TRENDS

Our examination of industry trends painted a positive picture of SaaS, with a rich application pipeline. Though the market is still relatively small, it is expected to grow up to 40 percent annually; as a result, some analysts expect that one quarter of all business software will be delivered using SaaS by 2011. In one survey of organizations using SaaS, over 90 percent said they were satisfied. Many software suppliers are creating SaaS solutions based on their traditional offerings; this will create additional outsourcing options.

INTEL SAAS DEPLOYMENTS

Over time, Intel has experimented with or deployed a variety of SaaS applications, including four of the five most widely used SaaS categories: customer relationship management (CRM), human resources management systems, collaboration, and business expenses.

We surveyed 14 SaaS solutions that had been used at Intel. Of these, 11 were in production use: these included expense and time-card tools, online learning, hiring tools, and health benefits. Table 1 summarizes the characteristics of these applications.

Key findings of our survey included:

Uses and benefits. Intel's most successful SaaS projects have involved the delivery of non-critical capabilities: commoditized functions that do not contribute directly to Intel's competitive advantage. The biggest benefit has been that SaaS provides agility, with fast access to new functionality.

Table 1. Characteristics of Software as a Service (SaaS) Applications in Use at Intel

Application Categories	▪ Good candidates for SaaS are applications with industry-standard workflows, which do not involve intellectual property or sensitive data, such as human resources management, employee stock options, medical benefits, and expense reports.
Users	▪ The number of users varies—from participants in a small pilot project to a majority of Intel employees for several applications. ▪ About half of SaaS applications are used globally and half are specific to the United States.
Business Drivers	▪ Agility and time to market. ▪ Lower cost: No need to develop and maintain internal expertise for commoditized capabilities with industry-standard workflows.
Costing Model	▪ Most applications use subscription licensing, with a fee paid at regular intervals—yearly, quarterly, monthly. ▪ A few applications are priced per transaction.
Platform	▪ In about 30 percent of cases, suppliers provide a dedicated hardware platform for Intel's application; in the other 70 percent, the platform is shared. The application is typically not virtualized. ▪ In about half of the cases, suppliers provide a dedicated application instance for Intel; the rest use a standard multi-tenant shared instance.
Disaster Recovery	▪ There is a disaster recovery plan for most applications.
Security	▪ All SaaS solutions have undergone a security risk assessment. Most data has a relatively low security rating. ▪ Half of the SaaS applications use single sign-on (SSO); the rest use personal profiles. ▪ Data may be encrypted in transit and at rest. Native Web applications use HTTPS/SSL to protect important data during transmission.
Monitoring	▪ Typically, vendors monitor applications and infrastructure and send us selected or summary alerts.

A Cloud Computing Taxonomy

To promote the use of common definitions, Intel has developed a cloud computing taxonomy. We referenced existing works of cloud taxonomy and used them as input to develop our own. The taxonomy includes several established categories of cloud computing service, as shown in Figure 1.

Elements of the taxonomy include:

- **Software as a service (SaaS).** On-demand software applications. With SaaS, software applications are rented from a provider as opposed to purchasing them for enterprise installation and deployment. At the top of the pyramid, this is the most mature category of cloud service; a wide variety of applications are already available for enterprise use
- **Platform as a service (PaaS).** On-demand software development platforms.
- **Infrastructure as a service (IaaS).** On-demand computing infrastructure.
- **Cloud software.** Unique purchased/packaged software used to build and run cloud services.
- **Service as a service.** Horizontal service that is subscribed to and used as a component of SaaS, IaaS, or PaaS offerings. An example is a billing service.
- **Cloud client.** Client-centric services and run-time software for cloud execution.

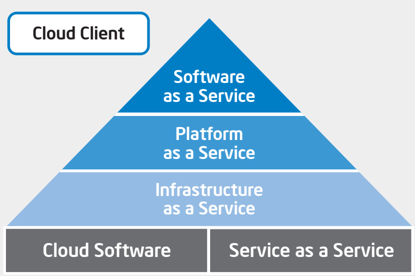

Figure 1. Intel adopted a cloud computing taxonomy using input from existing works.

We realized even more value when we outsourced the application-specific help desk along with the software.

We have also benefited from the expertise of outsourcing suppliers, which has enabled us to focus internal resources on other more critical tasks.

We have generally seen reduced costs from using SaaS solutions—however cost has historically been a lower priority than functionality. We like the ability to pay for only what we use, and we like solutions that could scale up and down based on demand.

Users generally have reported good experiences, especially in the areas of usability and responsiveness. In some cases, users had difficulty distinguishing SaaS from internally installed enterprise applications. We observed that Intel users are technically savvy and like a self-service approach, so they are comfortable with the SaaS model.

One innovative SaaS use has been to temporarily try new capabilities to inform our broader strategy and long-term plan. For example, we experimented with an on-demand CRM product with the intent of meeting immediate internal demand while we planned the deployment of an enterprise CRM solution. When the product did not meet Intel's needs, we were able to de-provision it quickly. This exercise provided insights that have influenced our current CRM direction.

Integration. The business process, application, and data integration required depends on the extent to which a SaaS application is tightly coupled with the Intel environment. We found that the key has been to evaluate the intersection of each SaaS solution with our existing business processes, systems, and data.

Ideally, we want a good fit between a SaaS application and our business process, with minimal need for configuration or customization. Where changes are required, we have favored configuration of SaaS applications over customization, but we have customized software in some cases where it is necessary to meet Intel business needs. We gathered best practices for testing and for troubleshooting—which is challenging when multiple parties, including an Internet service provider, are involved.

Security. With our existing SaaS applications, we have adopted a comprehensive approach to security. The rigor of our assessment was based primarily on the security rating of data; we assign these security ratings relative to the sensitivity and importance of the data. We requested evidence of protection level and encryption; with multi-tenant solutions, we probed how suppliers separate Intel's information from that of other companies. In some cases, we required a physical inspection of the provider premises and employee background checks.

Regulatory compliance. We identified a few areas as important for regulatory compliance. Employees' personal data must be protected, such as bank account information associated with expense reports. We also need to help ensure intellectual property protection; for example, we do not want to store sensitive documents in the cloud for potential access by controlled countries. We have to comply with local and national financial reporting regulations. We explicitly defined jurisdiction of data in contracts, including liability in the event of legal action.

Use-Case Model

As part of our SaaS architecture, we defined a use-case model that shows how the system behaves from a user-centered design perspective. Our model, shown in Figure 2, focuses primarily on IT-approved solutions for commoditized functions used by many Intel employees, though there is enough flexibility in the model to include some cases in which an Intel department or end-user sources their own solution.

There are three types of primary SaaS user roles within Intel.

- **IT users.** Primary IT user roles are applications administrators and SaaS specialists. The administrator is responsible for the decision to use

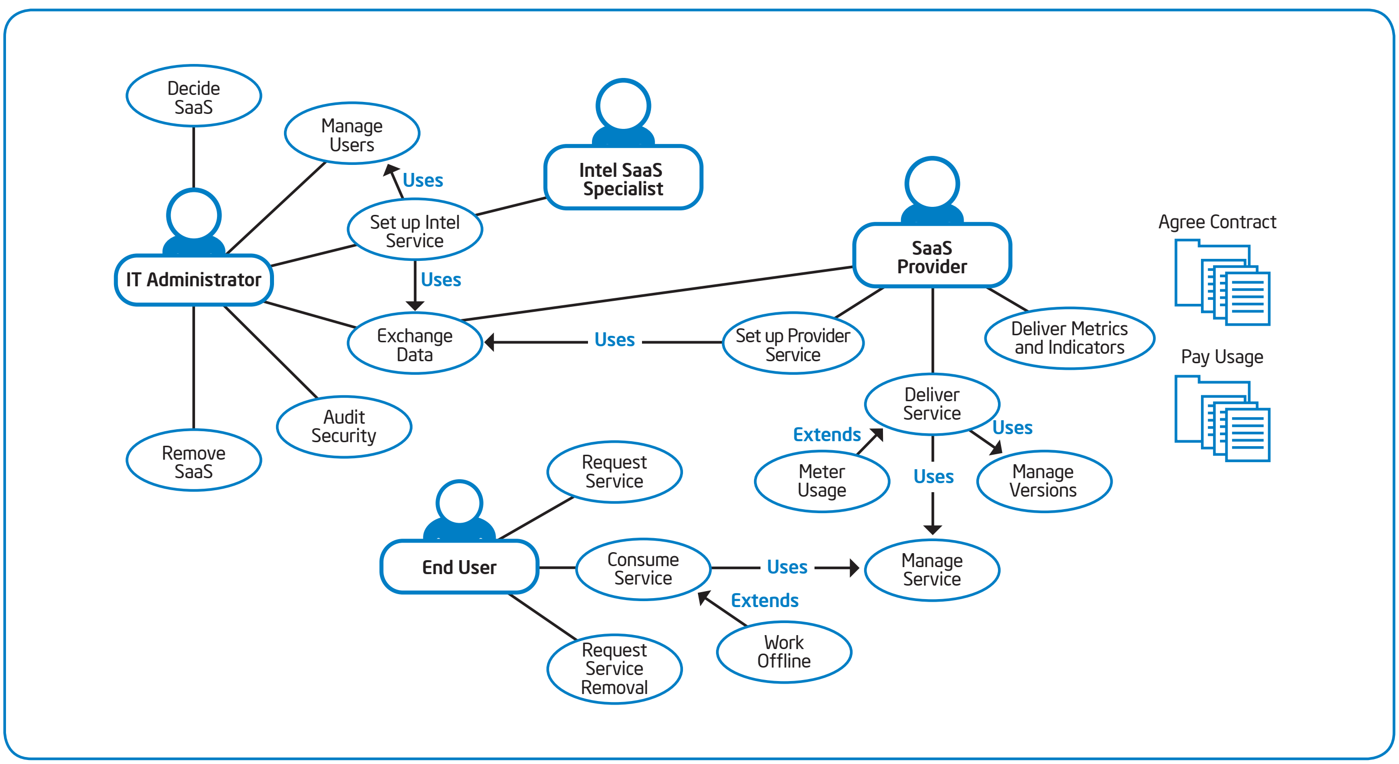

Figure 2. Software as a service (SaaS) architecture use-case model.

SaaS for a particular application and for any integration work needed to deliver the service within Intel. The SaaS specialist is the technical resource who delivers any personalization, programming, and customization for Intel.

- **End users.** Primary end users are individual workers at Intel who use SaaS applications for job-related activities. Workers are located within the enterprise or connected to Intel while traveling or working from home.
- **SaaS provider.** This is the external provider that delivers a software service over the Internet to Intel.

The model describes a typical SaaS solution in which back-end integration is required. It encompasses the entire service life cycle and includes use cases to define SaaS selection, initial setup at Intel and the SaaS provider, user consumption of the service, ongoing data exchange and administration, and service end of life.

Conceptual Architecture

The conceptual architecture is intended to represent a three- to five-year vision of SaaS architecture, free of implementation technology details, and to establish common capability definitions. The conceptual architecture depicts all the key capabilities required in a complete SaaS offering, the logical separation of capabilities into tiers, and the logical grouping of capabilities. We do not expect that individual SaaS applications will necessarily include every capability described in the conceptual architecture.

KEY FEATURES

A well-designed SaaS application has several key architectural features. It should be:

- **Multi-tenant efficient.** The design should support multiple tenants using a single instance of the application. The data must be segregated for each tenant.
- **Configurable.** The application can be configured to meet the needs of each tenant, using metadata and a metadata execution engine—also known as a business rules engine. Routine configuration changes should be possible without the need to coordinate downtime with other tenants.
- **Scalable.** Multi-tenant usage can result in millions of users. Applications should be designed from the ground up to scale up and scale out—and to be able to do this dynamically, on demand.

SAAS CAPABILITIES

Many capabilities make up the SaaS conceptual architecture. We group these into presentation, security, application, operations, and infrastructure categories, as shown in Figure 3. The following sections describe the most important capabilities.

Presentation

This includes all capabilities exposed to the user, such as:

- **Menu and navigation.** These provide access to the features and functionality within an application, organized in an intuitive way so that the user can select the desired function.
- **Reporting.** Application-specific predefined or ad-hoc reports.

Security

Security is one of the most important categories of SaaS capabilities, given that Intel's data and user accounts are typically hosted by the SaaS provider. We considered the following capabilities:

- **Identity and federation.** Identity uniquely identifies a user or another entity such as an Intel application or system. An example is a user name. Federation describes the function of enabling users in one domain to securely and seamlessly access data within another domain.
- **Authentication and single sign-on (SSO).** The process of identifying an individual, usually based on a user name and password. In the context of SaaS, this includes the ability to achieve SSO across multiple cloud applications and services.
- **Authorization and role-based access control.** After an identity has been confirmed, authorization is the process of giving individuals access to system objects based on their identities. Identities are usually assigned to roles for ease of managing access.
- **Entitlement.** The process of granting access to a specific resource. Tenants are usually responsible for maintaining their own user accounts using delegated administration.
- **Encryption.** Data may need to be encrypted in transit (between applications or between the layers within an application) and at rest (while stored).
- **Regulatory controls.** Tracking and reporting who accessed what, when, and why. It includes tracking access to application features and data, the security rating of the data, and the implementation of a data retention policy. It also includes identifying whether individuals are located in controlled countries.

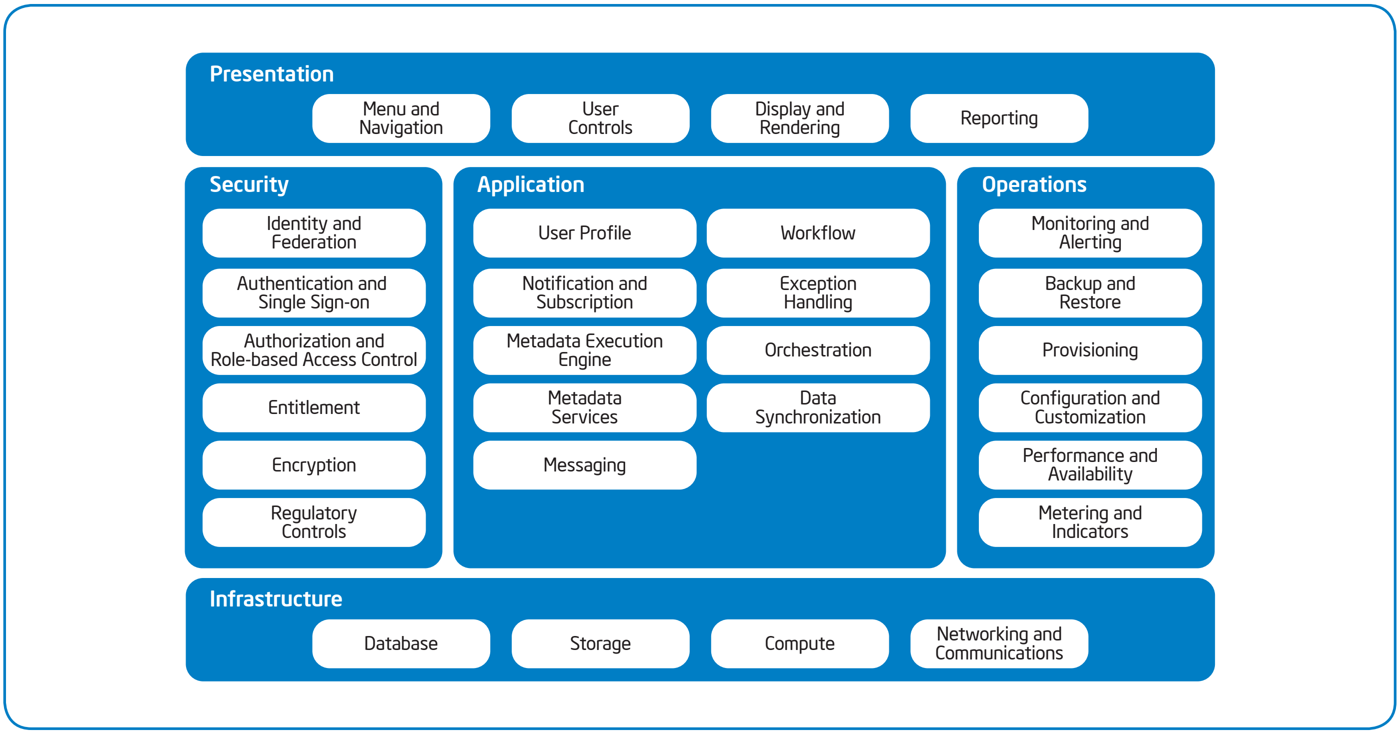

Figure 3. Intel IT's conceptual architecture for software as a service (SaaS) includes capabilities in five categories.

Application

These represent the typical business layer or middle tier of a SaaS application:

- **User profile.** The attributes and information that describe a user, such as name, e-mail address, and role.
- **Metadata execution engine.** Statements that define or constrain some aspect of the business. They are intended to assert business structure or to control or influence the behavior of the business.
- **Metadata services.** Information about which data is contained and exposed within an application and about how content is organized.
- **Workflow.** The defined series of user-based tasks within a process to produce a final outcome. An example is creating a purchase order.
- **Exception handling.** The process of raising and managing exceptions within an application. This includes how application errors are exposed to the user and how error messages are logged.
- **Orchestration.** The series of technical tasks performed within a process to produce a final outcome. An example is an extract, transform, and load sequence to move data between business applications.
- **Data synchronization.** The capabilities for synchronizing data held within the application with external data.

Operations

These are the capabilities needed to efficiently keep the SaaS application running:

- **Monitoring and alerting.** Polling application components, services, and infrastructure to detect failures. On detection, an alert is sent to the appropriate support group.
- **Performance and availability.** Performance describes how the application performs under load, both in terms of the number of users and the transaction volume. In the context of SaaS, this should allow applications to dynamically scale based on runtime usage and demand. Availability is a measure of how much of the time the application is available to users and is represented as a percentage.

- **Metering and indicators.** Tracking and reporting items specifically related to the service-level agreement, such as usage, availability, number of failures, and mean time to respond to and fix problems.

Infrastructure

The underlying technical capabilities required for storing data and moving it around the network:

- **Database.** In a multi-tenant data architecture, there could be one database per tenant or one database shared by multiple tenants with the data indexed by a specific tenant identification.
- **Compute.** The physical clients, servers, or virtual machines that execute code.

Reference Architecture

The purpose of the reference architecture is to provide a proven template solution that project teams can immediately apply to specific application domains. Accordingly, it includes only a subset of the capabilities described in the conceptual architecture and is more near-term in nature—one to two years. The reference architecture also provides a common vocabulary for discussing implementations; one goal is to increase the commonality between them.

Figure 4, on the next page, shows the high-level reference architecture for a typical SaaS offering at Intel. It includes summary views of data interchange, manageability, and security capabilities. We also developed more detailed architecture designs for each of these areas; key aspects of these are summarized in the following sections.

DATA INTERCHANGE

Intel SaaS project teams cite data interchange—moving employee data and other information between internal Intel systems and data stores hosted by the SaaS provider—as their biggest challenge. It's important to keep this data synchronized between internal and external systems, so data transfer may need to occur frequently, often on a scheduled basis.

The key challenge is locating the right version of the data, since data can be stored within the enterprise, in the cloud, or at both locations. Considerations include finding the master copy of data, searching for data, and governance.

Through our environmental scan, we discovered that various tools and designs are used to exchange SaaS data today, with no common architecture across all the implementations. This highlights the need to standardize on a single data interchange reference architecture.

We've identified two types of data interchange: asynchronous and synchronous. An asynchronous, or batch, interchange is typically used for back-end data exchange. For example, a SaaS expense report application needs to know about the management structure to enable management approval of travel expenses. This requires employee data to be periodically copied from the enterprise to the expense report application.

In contrast, synchronous, or real-time, interchange involves data that is dynamically retrieved in real time directly from its source. Today, there is limited use of real-time exchange for enterprise data, but we anticipate increased use in the future.

SECURITY

SaaS providers must comply with a number of security policies. We have done a good job of assessing providers in advance of implementation to help ensure they meet our requirements, but we will continue to move cautiously.

Additional work is required to qualify an externally hosted SaaS solution, so it is important to identify whether SaaS is an option early in the life cycle of an application implementation project; the additional security review and requirements may affect the overall agility and viability of the project.

Intel's security controls tend to be more mature than those of SaaS suppliers, and we must consider complex legal and regulatory requirements. Providers must be able to explain how jurisdiction of data is maintained. Together with the provider, we must be prepared to respond to e-Discovery and legal notices. We must also address privacy concerns, comply with export restrictions that cover access from controlled countries, be prepared to satisfy audit requirements, and understand how the supplier verifies that old data is destroyed.

From our environmental scan of existing deployments, we identified several other key elements needed for a successful project:

- Classification of the data by an IP attorney.
- Rigorous due diligence to help ensure the required controls are included in the contract.
- Completion of an information security risk assessment.
- Protection of data in transit and at rest.
- Making sure that suppliers provide satisfactory disaster recovery and business continuity plans.

For our users, it would be ideal to be able to achieve SSO or reduced sign-on to avoid continually re-authenticating for each application. In the future, we'd like to see this implemented with a federated identity approach using tokens that the SaaS provider can decrypt and read. This would enable users to log on to the Intel network and immediately have access to all their applications, without having to maintain individual profiles for each SaaS provider.

MANAGEABILITY

To date, automated manageability has not been a priority for us because we have had relatively few applications and they have generally not been business-critical. Intel relies largely on user feedback and SaaS provider data to help ensure the providers are meeting contractual obligations and addressing inadequacies.

Today, Intel users who are experiencing problems typically call their local Intel service desk. It is then the support agent's responsibility to escalate the issue to the SaaS provider's service desk.

As we prepare for wider deployment of more-critical applications, the ability to automatically acquire reliable and complete manageability data will become increasingly important.

We plan to work with standards bodies to develop verifiable manageability standards and certifications for use by service providers. This will provide organizations consuming SaaS applications with a common set of metrics, which will eliminate many of the initial validation steps currently required.

How much manageability data we will require from SaaS providers is still an open question. Our goal is to minimize introspection into incidents, focusing more on application performance, reliability, and common tracking of requirements.

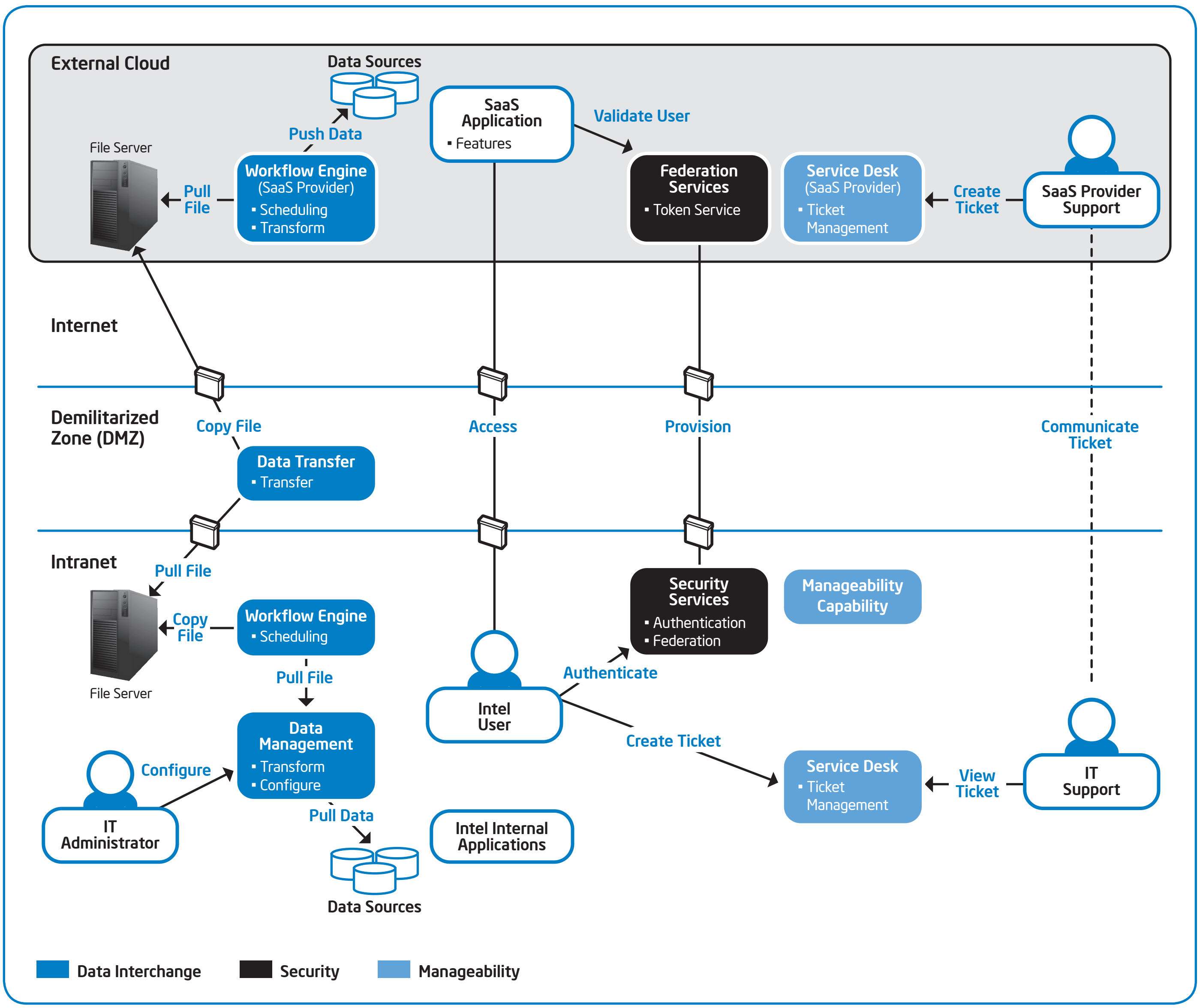

Figure 4. Intel IT's summary view of software as a service (SaaS) reference architecture.

CONCLUSION

The success of SaaS applications at Intel to date, together with our industry analysis, suggests that adoption will continue to grow. Our goal is that our SaaS architecture enables Intel's use of SaaS to progress from organic growth to prescriptive deployment, with the reference architecture helping to drive consistent designs, quickly, for new SaaS projects.

We are continuing to add to our architecture capabilities for exchanging information, managing solutions, and increasing security.

The architecture provides a critical level of consistency; building and deploying solutions that are similar in design enables Intel to reuse capabilities and reduce the amount of effort and time required for each project.

For the future, we are targeting governance and auditing as two areas for additional design consideration. Using our SaaS architecture as a guide, we will continue to work with our suppliers to standardize on capabilities that will enable faster and more cost-effective deployment of business solutions that are easier to integrate, more manageable, and highly secure.

CONTRIBUTORS

William Giard
Thiru Thangarathinam
Jay Hahn-Steichen
Stacy Purcell
All members of the Intel IT SaaS team

ACRONYMS

CRM	customer relationship management
IaaS	infrastructure as a service
PaaS	platform as a service
SSO	single sign-on
SaaS	software as a service

For more straight talk on current topics from Intel's IT leaders, visit www.intel.com/it.

Printed in USA
1009/KC/KC/PDF

Please Recycle
322460-001US