Cloud Customer Architecture for Web Application Hosting, Version 2.0

Executive Overview

This paper describes vendor neutral best practices for hosting web applications using cloud computing. The architectural elements described in the document are needed to instantiate a web application hosting environment using private, public, or hybrid cloud deployment models.

At a high level, web application hosting supports server applications which deliver web pages containing static and dynamic content over HTTP or HTTPS. The static content is typically represented by "boilerplate text" on a web page and more specialized content held in files such as images, videos, sound clips, and PDF documents. Dynamic content is typically built in response to a specific request from the client, based on content in the request and content derived from a database connected to the web application.

The core component for hosting web applications is the web application server, but to produce a secure, reliable, high performance architecture a number of other components may be required, such as firewalls, load balancers, transformation and connectivity functionality, enterprise data, file repositories, content delivery networks, and robust security. In addition, lifecycle management, operations management, and governance need to be considered for these components. How these functions are accomplished will differ depending on where the components are deployed and how integration into management systems is supported.

When the cloud service is an Infrastructure as a Service (IaaS) offering, all of the elements of the architecture will need to be individually acquired or instantiated. In some cases, the IaaS cloud service provider is able to offer some of the elements in a ready-to-run form.

For the case where the cloud service is a Platform as a Service (PaaS) offering, it is often the case that many elements of the architecture are available as part of the offering and only configuration and deployment is required.

The cloud deployment model affects the locations of many of the components. For public cloud deployment, the elements are instantiated in the public cloud. For private cloud deployment, the components are instantiated within the private cloud, either on-premises or within a privately managed environment made available by a cloud service provider. For hybrid cloud deployment, there is an element of choice of where to locate each component, with the choice typically governed by security, data residency regulations, and performance considerations.

Please refer to the CSCC's *Practical Guide to Cloud Computing* [1] and *Security for Cloud Computing: 10 Steps to Ensure Success* [2] for a thorough discussion on deployment and security considerations for cloud computing including recommendations on how best to address specific requirements.

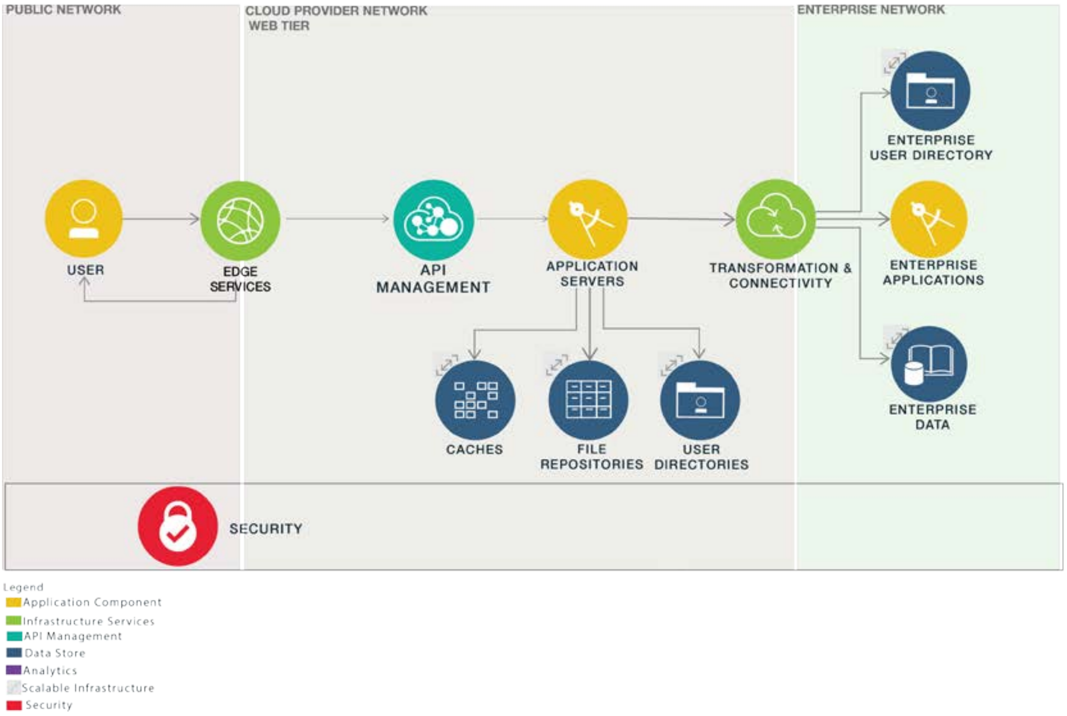


Figure 1: Web Application Hosting Cloud Architecture

Web application hosting is ubiquitous in the computing world and represents a generic pattern that can be applied in many situations. Cloud computing and cloud services are often considered for both existing and new web application hosting environments. This is in part driven by the frequency that web applications are required. It also occurs because cloud elasticity and scalability naturally lends itself to the needs of a web application hosting environment.

The following section describes the various components in detail.

Components

Public Network Components

User – Users can interact with the web application from a variety of devices and systems.

Edge Services – Edge services include network service capabilities needed to deliver content to web applications and its users through the Internet. These include:

Domain Name System
Content Delivery Network
Firewall
Load Balancer
EDGE SERVICES

- **DNS Server –** The Domain Name System (DNS) server maps the text URL (domain name) for a particular web resource to the TCP-IP address of the system or service that can deliver that resource to the client.
- **Content Delivery Network (CDN) –** Content Delivery Networks are geographically distributed systems of servers deployed to minimize the response time for serving resources to geographically distributed users, ensuring that content is highly available and is provided to users with minimum latency. Which servers are engaged will depend on server proximity to the user and where the content is stored or cached.
- **Firewall –** A Firewall is a system designed to control communication access to or from a system, aiming to permit only traffic meeting a set of policies or rules to proceed and blocking any traffic that does not meet these policies. Firewalls can be implemented as separate dedicated hardware, or as a component in other networking hardware such as a load-balancer or router or as integral software to an operating system.
- **Load Balancer –** Load Balancers distribute network or application traffic across many resources (such as computers, processors, storage, or network links) to maximize throughput, minimize response time, increase capacity, and increase reliability of applications. Load balancers can balance loads locally and globally. Considerations should be made to ensure that this component is highly available and is not a single point of failure.

Cloud Provider Network Components

Web Service Tier

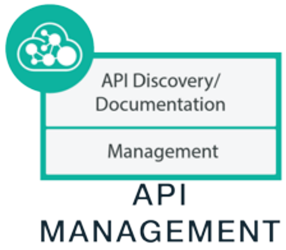
The provider cloud can host the web services tier which contains the program logic used to generate dynamic web content. This can involve retrieval of data from files, databases, HTTP-services, sensors, and other sources of data as well as programmatic generation of new data or information. Web servers and application servers can also be instantiated in a 3-tiered setup with separated, rather than integrated, web servers and application servers. In that case, there would be separate pools of web servers and application servers connected via load balancers. The application server would be responsible for accessing databases or other systems. Components include:

Web Application Servers
Cache
File Repository
User Directory
WEB SERVICE TIER

- **Web Application Servers –** Web Application Servers offer web server functionality and integrated application server functionality if it is needed. Web servers are systems that return resources (web content and images, for example) in response to an HTTP request and may be configured to handle requests for multiple IP addresses and/or domains. Web application servers may support clustering, pooling, and other high availability and scaling configurations including auto scaling – instantiating and removing application server instances as demand requires.

- **Cache –** Caches store information temporarily needed to fulfill a request by the web application server, including session data and other content. The purpose of the cache is to reduce the latency in responding to a request from a client.
- **File Repository –** File repositories are devices or applications that store information, data, etc. in the form of files. Access to the file repository generally includes the ability to store, retrieve, delete, and search the repository for a particular file. File repositories can use network storage to provide access to shared files.
- **User Directory –** User Directory contains user IDs and credentials needed to validate that the user is allowed to access the information or applications being requested in the Web servers and application servers. The directory can be accessed by web servers, applications servers, databases or any other elements used in the web application.

API Management – API management capabilities advertise the available services endpoints to which the application has access. It provides API discovery, catalogs, and connection of offered APIs to service implementations and management capabilities, such as API versioning. APIs and services are the contemporary foundation for transformation and much connectivity. The design and operation of these services should address eight key elements to assure a solid foundation for a reliable transformation and connectivity strategy: composition, security, deployment, access, governance, analytics, management, and scalability.



- **API Discovery/Documentation –** Provides the ability for mobile developers to find and use APIs securely.
- **Management –** Provides a management view into API usage by web applications and mobile apps using information from mobile gateway, backend, etc.

Transformation and Connectivity – The enterprise transformation and connectivity component enables secure connection to enterprise systems and the ability to filter, aggregate, or modify data or its format as it moves between web components (systems of engagement) and enterprise systems (typically, systems of record). Within the web application reference architecture, the transformation and connectivity component sits between the web and enterprise tiers. However, in a hybrid model, these lines might become blurred. The elements that comprise this architecture domain, listed below, were until very recently listed as separate tool types: Enterprise Application Integration (EAI), Data Integration (DI) and Extract, Transform, and Load (ETL). The Transformation and Connectivity component includes the following capabilities:

Enterprise
Security Conectivity
Transformations
Enterprise Data
Connectivity
TRANSFORMATION &
CONNECTIVITY

- **Enterprise Secure Connectivity –** Leverages security services to integrate with enterprise data security to authenticate and authorize access to enterprise systems.
- **Transformation –** Transforms data between enterprise systems.

- **Enterprise Data Connectivity –** Provides the ability for mobile components to connect securely to enterprise data. Examples include VPN and gateway tunnels.

Enterprise Network Components

Within enterprise networks, enterprises typically host a number of applications that deliver critical business solutions along with supporting infrastructure like data storage. Typically, applications will have sources of data that are extracted and integrated with services provided by the cloud service provider. Analysis is performed in the cloud computing environment, with output consumed by on-premises applications. Any data from enterprise applications can be sent to enterprise or departmental systems of record represented by the enterprise data components. Systems of record data have generally matured over time and are highly trusted.

Service Tier

Enterprise User Directory – Provides storage for and access to user information to support authentication, authorization, or profile data. Security services and edge services use this to manage access to the enterprise network, enterprise services, or enterprise specific cloud provider services.

Enterprise Data – Enterprise Data includes metadata about the data as well as systems of record for enterprise applications. Because of compliance requirements for localized data storage, the use of distributed database management tools that accommodate hybrid cloud architectures is essential to handling global user bases. Many types of enterprise data play a role in a web application hosting design. These include:

Reference Data
Master Data
Transactional Data
Application Data
Log Data
Enterprise Content Data
Historical Data
ENTERPRISE DATA

- **Reference Data –** Reference data provides the standard context for collected data. In some instances, Reference Data and Master Data are one in the same.
- **Master Data –** These repositories can be updated with the output of applications, enterprise applications, and analytics to assist with subsequent data transformation, enrichment, and correlation.
- **Transactional Data –** Data about or from business interactions that adhere to a sequence or related processes (such as financial, logistical, or other process). This data can come from reference data, master data repositories, and distributed data storage.
- **Application Data –** Data used by or produced by business solutions and enterprise applications functionally or operationally. Frequently the data has been improved or augmented to add value and drive insight.
- **Log Data –** Data aggregated from log files from enterprise applications, sensors, infrastructure, security, governance, and service providers.
- **Enterprise Content Data –** Data, frequently object files, to support any enterprise applications or B2B or B2C content delivery on a large scale.
- **Historical Data –** Data from past analytics and enterprise applications and systems. Use of cloud-based storage for archived data reduces storage costs and expedites use of analytics as a service and mining data for new insights.

Enterprise Applications – Enterprise applications can consume cloud provider data and analytics to produce results that address business goals and objectives. Enterprise applications can be updated from enterprise data or the web applications, or they can provide input and content for enterprise data or web applications. Applications might include:

Customer Experience
New Business Models
Financial Performance
Risk
IT Economics
Operations & Fraud
ENTERPRISE APPLICATIONS

- **Customer Experience –** Customer-facing cloud systems can be a primary system of engagement that drives new business and helps service existing clients with lower initial cost.
- **New Business Models –** Alternative business models that focus on low cost, fast response, and great interactions are all examples of opportunities driven by cloud solutions.
- **Financial Performance –** The office of finance should become more efficient as data is consolidated and reported faster and easier than in the past.
- **Risk –** Having more data available across a wider domain means that risk analytics are more effective. Elastic resource management means more processing power is available in times of heightened threat.
- **IT Economics –** IT operations are streamlined as capital expenditures are reduced while performance and features are improved by cloud deployments.
- **Operations and Fraud –** Cloud solutions can provide faster access to more data allowing for more accurate analytics that flag suspicious activity and offer remediation in a timely manner.

Security Components

Security for web application hosting addresses fundamental business needs of security such as:

- Right people having access to the cloud web applications and their data (Confidentiality)
- The data of business users are intact and not tampered (Integrity)
- Availability / uptime of cloud web applications despite many security threats (Availability)
- Help address industry and regulatory compliance needs (Compliance)

Security capabilities to address business needs include:

Identity & Access Management
Data & Application Protection
Security Intelligence
SECURITY

Identity & Access Management – Capabilities to identify and authorize the user providing role-based access to cloud web applications. It also enables single sign-on, user lifecycle management, and audit logging. The user types and their levels of access for cloud web applications need to be managed. This could include business users (customer, vendor, 3rd party, staff users), or IT users (administrators, privileged users, application users). Identity and access management could leverage the enterprise user directory from the service tier.

Data and Application Protection – Capabilities that help identify vulnerabilities and prevent attacks targeting sensitive data. It provides protection to cloud web applications against many malicious threats right from the beginning of the development cycle. In addition, it monitors privileged access to sensitive data. It also protects integrity of sensitive data in transit and at rest and provides network isolation. Firewalls in the public network component tier help protect the network level flows to application and data.

Security Intelligence – Capabilities to monitor the cloud web application for security breaches to provide visibility. It provides actionable intelligence to detect and defend against threats using event and log analysis that feeds to a corporate incident management system. Security reports support regulatory compliance of the cloud web application.

Security and security management applies across the cloud lifecycle - design, development, deployment and ongoing maintenance. Security governance is an integral part of security management.

The Complete Picture

Figure 2 provides a more detailed view of components, subcomponents, and relationships in a cloud-based web application hosting architecture.

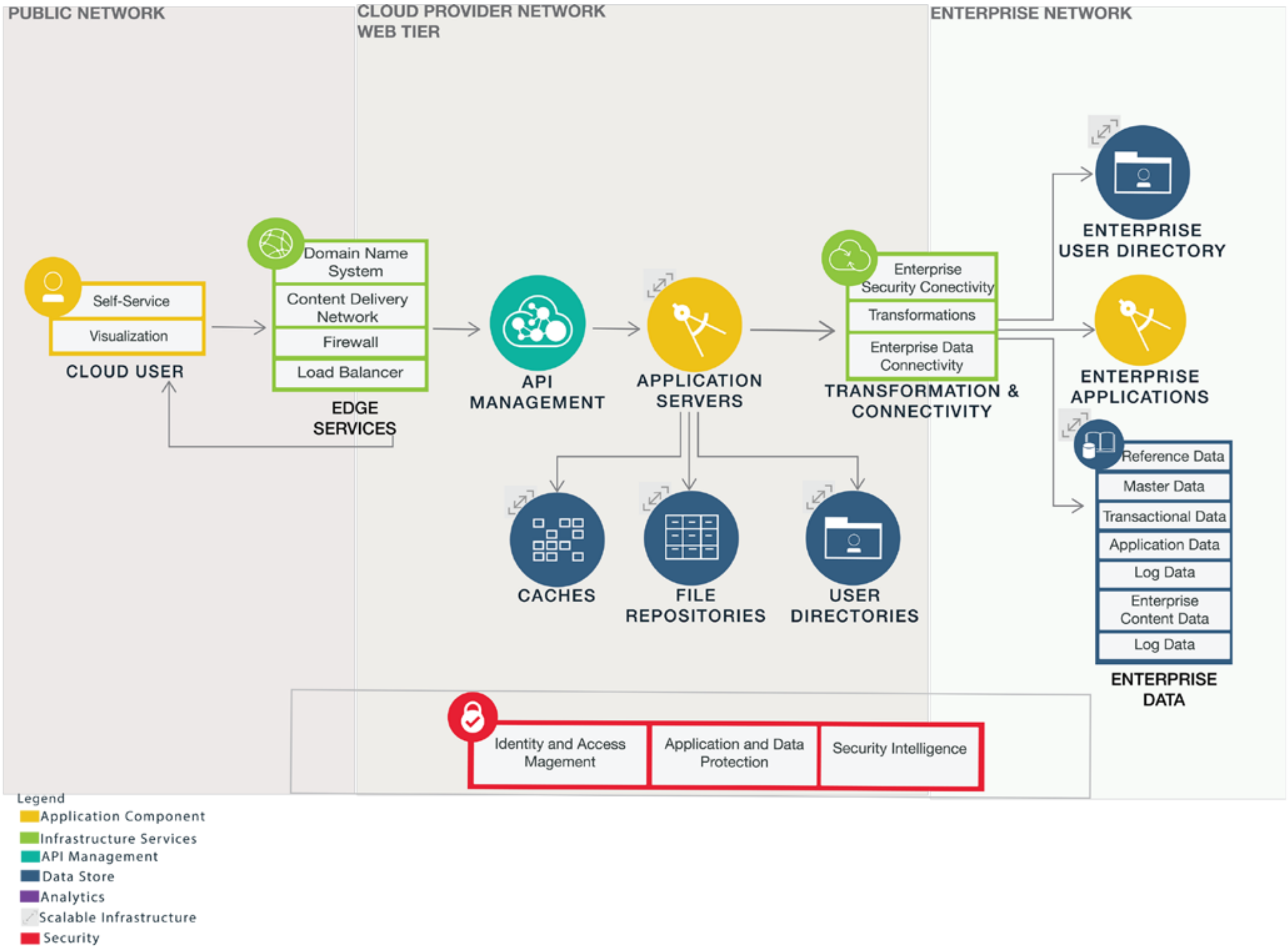


Figure 2: Detailed Components Diagram

Runtime Flow

Figure 3 illustrates a general purpose flow for the web application hosting architecture.

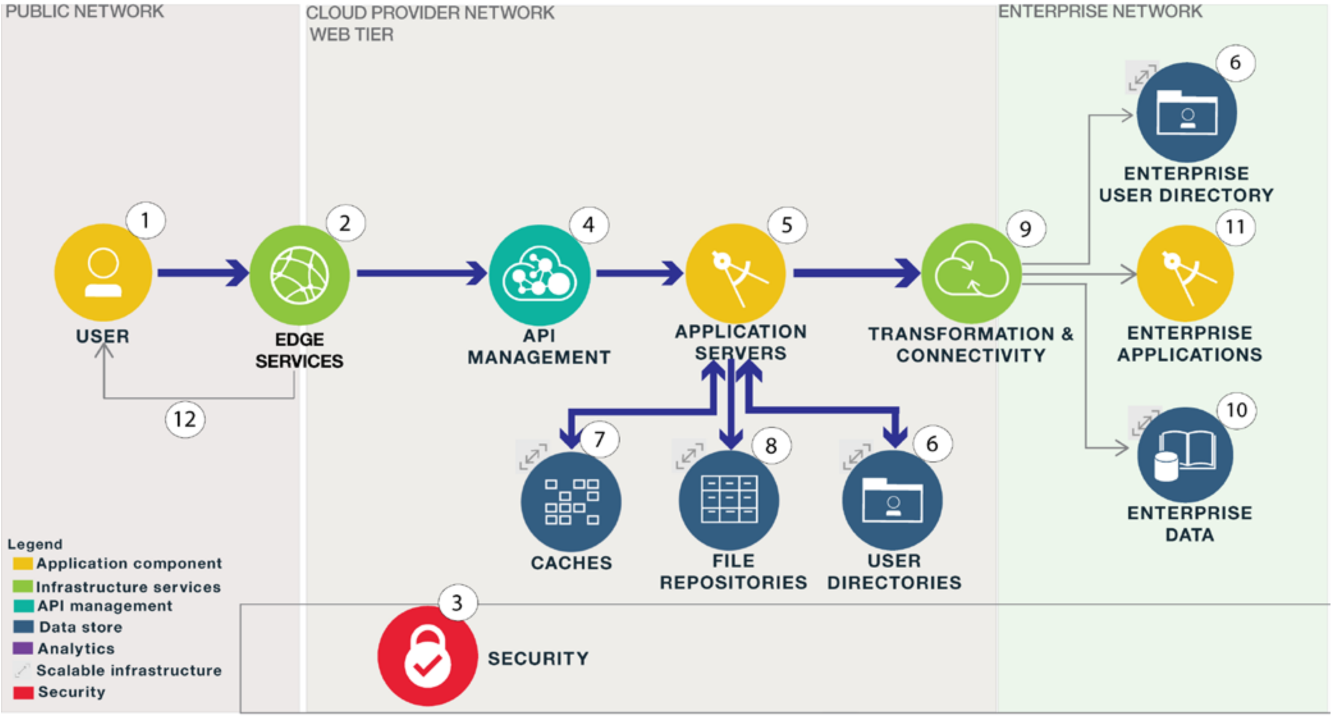


Figure 3: Flow for General Purpose Web Application Hosting

General flow includes:

1. A user agent (or user) sends a request to a specified URL.
2. Edge Services receives the request – Edge services consist of a group of services that handle the request and get it to the right destination. These include: the domain name server, the CDN server, the firewall, and the load balancers. Often an API manager is added to find the right application once the request is inside the network. Every request going to or from the network goes through the firewall.
    a. Domain Name Server (DNS) –The domain portion of the URL is resolved into an IP address via the Domain Name Service (DNS). This IP address may actually be the IP address of a CDN server, load-balancer, firewall, or proxy service in-front of the actual web application server that will satisfy the request.
    b. The CDN server determines if any of the requested content is in the CDN storage network. If the CDN server cannot satisfy the request, then the request is sent to the firewall.

   c. If the CDN server is able to satisfy the request leveraging content in closest proximity to the user, then the CDN responds to the request by returning that content. The user's browser retrieves and displays the returned content.
   d. If the CDN cannot satisfy the request, the message is passed to the firewall and then the load balancers. Both of these will use security services.
   e. Firewall **–** The firewall evaluates the packets that form the request and allows only those packets which meet the rules of the firewall to continue forward to the load balancer. Typical rules might only pass incoming HTTP and HTTPS packets destined for ports 80 and 443. Firewalls often have two sets of rules, one for filtering inbound traffic into the firewall and one for filtering outbound traffic going from the firewall. Generally, DNS resolution for internal requests is typically done using a private DNS server rather than a public DNS server.
   f. Load Balancers – The load balancer sends the request to a specific web application server in a pool of web application servers. The decision is made using a random or 'round robin' algorithm, or some other method. For example, it might pick the server currently doing the least amount of work (least load). If the packet is associated with a web-session, the load-balancer may direct the message to the server that most recently handled a request in the same session (stickiness). Load balancers can direct requests by processing sophisticated rules, using systems and business policies, current and historic performance, as well as resource usage and availability in the underlying VMs or systems.
3. Security **–** Security is enabled across multiple layers through a defense in depth approach. Cloud web applications have their access provided to the right users and roles through identity and access management. The web applications are protected from threats (such as cross site scripting, SQL injection attacks, and more) starting at the beginning of the development cycle. The application stack is further isolated at the network level into multiple network segments or VLANs. The sensitive data is protected from end users and privileged users. Continuous monitoring of threats and log analysis in the solution provide visibility and actionable intelligence. Logs are used for audit and compliance reports.
4. API Manager – The API manager receives the request and determines which services or applications in the applications server should be invoked and determines if that user has the appropriate authority.
5. Web Application Servers – The web application server returns a resource (normally some form of web content) based on the user's request. Based on the request, the web server retrieves the static content by accessing the file system or invokes a program or service to generate dynamic requested content.
6. Before any processing is done, the web application server may invoke the user directory to authenticate the user and validate permission rights to perform the request. Typically this is done as a part of a login process, which establishes a session used for a series of requests. The user directory may use security services and the enterprise user directory in the enterprise.
7. The web application server determines if the request can be satisfied by a local cache. If so, the appropriate content and associated data is returned to the user.

8. File repositories, like caches, store and managed data that can be requested by the application server. Caches and file repositories return data through the firewall (#2). If application logic must be invoked (by the application server) then retrieval of data from files, databases (#10), web-services, sensors, and other sources of data as well as programmatic generation of new data or information may be required.
9. Transformation and Connectivity takes the messages and data intended to be stored in the enterprises database and completes any necessary transformation from web formats to data base formats and ensures secure reliable messaging is used appropriately.
10. Enterprise Data – The web application server may need to access a database to query data in order to generate the requested response. That data may be accessed directly or may require transformation in order to be utilized by the application. Enterprise data includes logs and data bases to enable analytics.
11. Enterprise Application – The enterprise applications may use data used by the web application as well as logs and context data for analytics. If the web application updates the data then enterprise applications may process those changes.
12. When the web application server completes its tasks, the resulting content is delivered back through the firewall (#2) which passes the content to the user's browser.

The above describes a general purpose scenario. Web application flows may change depending on the openness of the application. For example, some applications require no identity management – meaning that the application is available to any user. In that case, portions of step #3 can be eliminated.

Deployment Considerations

Deployment of the components for web application hosting depends on the capabilities of the cloud service(s) which are chosen. These architectural decisions are rarely certain and many factors must be considered before drawing a conclusion. The guidance provided below is intended to provide a starting point for the issues that must be considered in making a selection. Each situation will vary.

Consideration	**Factors**	**Recommendations**
Cloud Service Category: Determine ideal service category to use (PaaS, SaaS, IaaS)	Skill level, technical debt, culture, ability and willingness to adopt changes	Lean toward PaaS if you have a new custom workload with little existing technical investments and an innovative culture. Lean towards IaaS if you have many existing workloads that are unlikely to change and a conservative, risk adverse culture. SaaS can be an ideal option for both conservative and innovative cultures, if the SaaS solution closely satisfies requirements.
Deployment Model: Determine ideal model (On Premises, Off Premises, Hybrid)	Data sovereignty and data location, cost, capacity estimates, security, performance and legal and local regulations	Lean towards on-premises if you have large diversified workloads that can be supported in a small number of data centers. Lean towards off-premises if you do not own, or want to own your own data center, need multiple data centers spread around the world, or do not have enough workload volume to drive

		efficiencies. Lean toward hybrid deployments if you have unpredictable capacity and elasticity requirements. Legal, regulatory, and performance requirements will have a significant impact on workload placement.
Connectivity & Network: Identify optimal network configuration	Performance, IP, security and cost.	Lean toward a software defined network if you have expertise and skill level to support it.
Data: Determine ideal data placement	Data sovereignty, security, industry standards, performance	The first consideration in data placement is regulatory requirements that may impact data residency or sovereignty. After determining data residency limitations, decide whether co-locating the data with the application is necessary for performance reasons. Then decide if it's best to push the data to the application or the application to the data based on the amount of data and the application characteristics.
Development: Select development tools and automation solutions	Integration with other services, performance, ease of use, developer productivity, development methodology, runtime support	Lean toward a PaaS set of tools if your organizational culture values innovation and time to market. Lean toward a more traditional development toolset for organizations that are conservative or have a desire to simplify and reduce the total number of tools supported. Automation will help either type of organization and should be encouraged.
Migration: Determine if existing applications can and should moved to the cloud, refactored, or redesigned	Skills, performance, cost, time to market, integration requirements	Lean towards a "lift and shift" approach if the application is stable, non-critical, and would not benefit from revision. Lean towards redesign if the application is likely to evolve in the near future.
Integration: Determine the best model for integration in hybrid deployments	Security, performance, cost, flexibility, acceptance of an API economy	Lean toward an API-based integration approach if the organization wants to coexist with partners as part of a larger ecosystem or make services more assessable internally or externally. Lean toward enterprise integration when there are complex transformations in the data or the service is not easily simplified into an API.

Common Deployment Decisions

Deployment of the components for web application hosting depends on the capabilities of the cloud service(s) which are chosen. DNS and CDN usually live in the public network – these are typically purchased as services from a suitable provider.

For IaaS:

- The firewall and load balancer are deployed in the cloud service. Many cloud service providers have firewall services available, either in the form of specialized hardware devices or in the form of dedicated systems running suitable software. Multiple redundant firewalls can be deployed to avoid a single point of failure, each handling a unique IP address, where the capabilities of DNS servers to map one URL to multiple IP addresses can be used. The load balancer can be run on one or more separate servers within the cloud service, or it can run on the same system(s) as the firewall.
- It is typical for the web application servers to be run in multiple instances, all serving the same web pages. This can be to deal with the expected throughput of requests and also to provide resilience against failure of a single instance. The location of instances can be a consideration, where physically remote instances can help deal with problems that affect a single data center.
- The user directory, the cache, and the file repository are run as instance VMs in the IaaS service, with redundancy and failover for each.
- The databases are run as VMs, with multiple instances and replicated data stores for capacity and resilience purposes.

For PaaS:

- The firewall and load balancer are part of the platform and simply require configuration.
- The web application servers are also part of the platform - the application code must be loaded onto them, but the running of multiple instances is handled by the platform and all that is required is configuration of the number and location(s) to use.
- PaaS usually supplies the database as a service, and it is common for the database service to provide replication of data and scalability of instances.

Regardless of where components are deployed – public, private or hybrid – lifecycle, operations, security, and governance requirements need to be considered and addressed. Where components are deployed will strongly affect how management and governance are done. Private deployments may be able to use existing internal management and governance tools if they have access to the cloud infrastructure. For public, hybrid, and externally hosted private deployments, lifecycle operations – instantiate, initiate, terminate – for components outside the firewall need to be negotiated with the hosting parties.

Similarly, operational monitoring and management capabilities – for gathering metrics, checking SLAs, status, notifications, and negotiating changes in capacity – require that access to the related cloud service administrative interfaces needs to be obtained and support for them should be added appropriately to existing management tools. This may include integrating data, information, tools, and processes from multiple sources into common interfaces, reports, and automation tools for efficient and scalable operations.

Governance and compliance processes will need to accommodate the change in control and risk over any externally hosted components, especially where changes are controlled by the cloud service provider. Optimally, lifecycle management solutions should integrate across deployment models and provide a common, integrated context that enables management of release, change, security, SLAs, and problem diagnosis.

References

[1] Cloud Standards Customer Council (2014). *Practical Guide to Cloud Computing*. http://www.cloud-council.org/resource-hub.htm#practical-guide-to-cloud-computing-v2

[2] Cloud Standards Customer Council (2015). *Security for Cloud Computing: 10 Steps to Ensure Success*. http://www.cloud-council.org/resource-hub.htm#security-for-cloud-computing-10-steps-to-ensure-success

Acknowledgements

Major contributors to this whitepaper are: Heather Kreger, Mary Mt. Pleasant, John Bell (Ajon Technologies), Britni Lonesome, Karolyn Schalk, Karthik Sivakumar, and Mike Edwards (IBM).