## Contents

1. Introduction  4
   - Building a World-Class SaaS Operation  4
   - Focus on Data Center Operations  6

2. Architect  7
   - How SaaS Architecture Affects Service Delivery  7
   - Operability Planning  8
   - Best practices for effective service delivery architecture  8
   - Data Management Planning  8
   - Availability Planning  9
   - Capacity Planning  10
   - Considerations for Delivery Partnerships  10

3. Secure  11
   - Categories of SaaS Security Attacks  12
   - SaaS Operational Countermeasures  14
   - Access Controls  14
   - Private Connectivity  14
   - Security Partnerships  15

4. Comply  16
   - Compliance authorities  17
   - Data sovereignty and government control  18
   - Overlapping compliance areas  19
5. Operate 20
   Goals and Functions of SaaS Operations 20
   A Roadmap for Operational Excellence 23

6. Summary 24

Service Delivery Checklist 25
Glossary of SaaS Terms 27
1. Introduction

Software as a Service (SaaS) is a business model where products are served over a network and billed on a subscription or on a consumption basis. SaaS promises full-featured applications that are delivered and managed by a team of domain experts. SaaS customers expect the highest levels of security, performance, and compliance. They also demand high availability, full disaster recovery, and consistently high performance. To profitably attract and retain customers, SaaS vendors must excel at SaaS delivery. This guidebook explains how to do just that.

The rapid growth of SaaS has led to increased visibility lately with several high-profile security breaches that have compromised some major SaaS providers. These incidents threaten the reputations of the companies involved. And study after study cites ever-increasing cybersecurity threats. For this reason, all SaaS organizations need deep and evolving security expertise to safely deliver their solutions.

In addition to basic data security, SaaS applications are expected to comply with all relevant industry standards and audit requirements. These typically cover areas such as data sovereignty; safeguarding of personally identifiable information; prevention and detection of fraud; transaction traceability; and many other topics.

Successful SaaS vendors may experience rapid growth and fluctuating loads. They must be able to respond to ever-changing capacity requirements while maintaining the performance that their customers demand. SaaS providers must accommodate usage spikes without service interruption or performance degradation. They must also deliver a high-quality experience for an increasingly global base of users.

SaaS solution delivery must be cost-effective. This requires careful analysis of the financial trade-offs such as build-versus-buy as well as when to engage in strategic alliances and partnerships. Automating and optimizing key operational tasks is critical to delivering SaaS profitability.

Building a World-Class SaaS Operation

IT operations is an inherently integrative endeavor. It combines aspects of systems design, infrastructure hosting, tools, third-party software, and operational personnel. IT processes serve to bind these components into a cohesive framework. In addition to this guidebook, process frameworks such as Information Technology Infrastructure Library (ITIL), Information Technology Service Management (ITSM), and Capability Maturity Model Integration (CMMI) can help to guide organizations and to assess their operational progress.
The key steps in building operational excellence are described next. They will be explored in greater detail in subsequent sections.

1. **Architect:** Successful operations begin with the design of systems for performance, flexibility, availability, and cost efficiency.

2. **Secure:** Both physical and information security must be a central focus throughout systems design, implementation, and operation.

3. **Comply:** Best-in-class SaaS operations adhere with industry and government standards. Examples include PCI-DSS, SSAE-16 SOC-2, ISO 27001, and many more.

4. **Operate:** To deliver reliable and high performance SaaS solutions, providers must have a relentless operations focus.
Data Center Architecture:
Mission-critical operations must be built on a solid foundation, making high-quality data centers a must for SaaS providers. Locating data close to users means implementing a distributed application architecture. This often means hosting your operations in multiple locations. Consider augmenting owned facilities with hosting partners who can help you grow and that enable business agility.

Physical security:
Multi-tenant data centers are inherently more secure than single-occupant facilities. They’re designed literally from the ground up to provide physical security and isolation of each tenant’s systems. This includes strict visitor authentication, highly controlled access, continuous surveillance of all personnel on the site, and a detailed audit trail of who comes and goes.

Infrastructure Compliance:
Data centers are a key focus area for operational audits and compliance. Facilities should be certified to comply with industry standards such as SSAE16 SOC-1 Type II and/or ISO 27001. Use of certified facilities simplifies compliance with standards such as HIPAA and PCI. Many countries have data sovereignty and privacy requirements that necessitate local operations. Look for a global service delivery partner with the knowledge and resources to help meet these requirements.

High-Availability Operations:
SaaS users have very little tolerance for downtime. Yet most enterprise data centers do not meet the same level of quality and availability as those run by colocation providers. Be sure to deploy in facilities with highly redundant power and cooling infrastructure. Also, be sure you have a choice of networks to optimize traffic performance and cost, and to avoid vendor lock-in.
2. Architect

Systems architecture establishes a formal description and specification of the application. This is a critical first step in planning for SaaS delivery.

How SaaS Architecture Affects Service Delivery

A myriad of architectural details affect SaaS operations. Factors such as data location and flows, encryption at rest and in transit, application performance requirements, and load variability all have major implications for SaaS delivery.

Service delivery requirements have a major impact on infrastructure design. These may take the form of informal goals, or they may exist in explicit customer Service Level Agreements (SLA). They may also vary by application module and target market.

It is vital for SaaS companies to establish clear competitive differentiators to distinguish themselves from similar application providers. This will help attract customers and avoid competition based solely on price. In terms of service delivery, key differentiators may include application uptime, performance, ability to quickly scale both up and down, and quality of support. Service architecture may also be influenced by regulatory requirements and conformance with established standards.
Operability Planning

Successful SaaS delivery depends on close cooperation between application development and service operations teams. The growth of the DevOps role illustrates the importance of this partnership. It is not sufficient for application developers to create a specification and simply “throw it over the wall” to the service delivery team. Application design should explicitly include support for monitoring, management, and maintenance. Collaboration between design and delivery teams reduces operational costs and makes application development, release, and delivery more efficient and reliable.

Resource Isolation and Multi-Tenancy

Efficient SaaS operations should always be architected for multi-tenancy unless prohibited by performance or security constraints. This allows for more efficient utilization of resources and personnel, thereby reducing costs. With multi-tenancy, customers can potentially share:

- Physical hardware
- Virtual machines (VMs)
- Networks
- Monitoring tools
- Databases

While multi-tenancy has benefits, it introduces security and complexity challenges. SaaS architecture must address these challenges to ensure customer data isolation, high application availability, and avoidance of performance impact due to “noisy neighbors”.

Best practices for effective service delivery architecture

- Architecture starts with defining the required user experience. Establish objective performance goals early in the design process. Then make sure your service delivery infrastructure is capable of meeting those goals.

- Even the most efficiently-coded application cannot overcome physics. Deploying application infrastructure and data close to your end users can do far more than code tuning to improve performance and user satisfaction.

- Avoid vendor lock-in. Choosing the most capable provider(s) in each market where you operate results in better performance and lower costs. Deploy your network edge in facilities that offer a wide choice of providers.

Data Management Planning

Perhaps the most challenging architectural decision is where to locate data. Distributing data close to end users leads to better application performance. It also enables compliance with data sovereignty requirements. But it can add considerable complexity and lead to reporting and data protection challenges.

Distributed data platforms, including NoSQL platforms like MongoDB, Apache Cassandra, and others help to mitigate some of these concerns, but they lack some of the consistency features of traditional relational database management systems. Eventual consistency may be sufficient for social networks, but it will not work in most transactional environments. Recent developments of so-called NewSQL databases, based on Google’s Spanner model, can help to bridge this gap.
Availability Planning

Another critical consideration for application architects is service resiliency. Assess the impact of each component on the overall system. A guiding principle is to avoid any single point of failure in the system. While this is simple in theory, implementation can be quite challenging. One approach is to use highly specialized fault tolerant hardware. Such systems are typically very expensive and don’t scale well. For SaaS environments, it is better to provide horizontal scale using commodity components. The application architect should assume that every component could fail, and should design the system so that no single component failure results in application downtime. This is probably best described in a seminal paper from Amazon Distinguished Engineer James Hamilton:¹

Expect failures. A component may crash or be stopped at any time. Dependent components might fail or be stopped at any time. There will be network failures. Disks will run out of space. Handle all failures gracefully.

Keep things simple. Complexity breeds problems. Simple things are easier to get right. Avoid unnecessary dependencies. Installation should be simple. Failures on one server should have no impact on the rest of the data center.

Automate everything. People make mistakes. People need sleep. People forget things. Automated processes are testable, fixable, and therefore ultimately much more reliable. Automate wherever possible.

¹ Hamilton, James, “On Designing and Deploying Internet-Scale Services”, Pp. 231-242 of the Proceedings of the 21st Large Installation System Administration Conference (LISA ’07)
Capacity Planning

SaaS service delivery infrastructure may need to adapt to rapid shifts in demand. It should be able to scale up and down in near real-time. The scaling requirements are highly dependent upon the type of SaaS product. For example, e-commerce applications may have large spikes during holiday seasons. Other systems may have more predictable demands and not require as much scalability.

SaaS architects should analyze anticipated and historical load patterns to identify highly variable versus relatively stable workloads. This may enable the implementation of an “own the base, rent the spike” approach to capacity management. With this model, stable and predictable workloads are run on cost-effective owned systems. Highly variable workloads can be shifted to a managed service provider or to a public cloud and consumed on-demand.

System hardware benchmarking is another important aspect of delivery architecture. Designers need to know the performance capabilities of each system component. Incorporating groups of systems into capacity “pods” is a popular way of providing incremental scale to meet growth and to provide redundancy. Additionally, newer storage technologies such as solid state disks (SSD) can profoundly impact the SaaS architecture to help deliver higher performance.

Finally, application architects need to pay particular attention to the design of their wide area network (WAN). The WAN is the glue that holds all other infrastructure pieces together. If it is undersized or unreliable, the performance of the overall system will suffer. Public Internet may not provide the security or quality of service (QoS) needed for robust service delivery. Consider high-capacity, secure, point-to-point backbone links between major locations to carry the bulk of back-end traffic.

Considerations for Delivery Partnerships

A trusted provider offers valuable expertise and the architectural, security and configuration guidance for constructing your service delivery environment. SaaS infrastructure has many parts. It is not always possible or practical for a SaaS provider to manage everything by themselves. Selective use of delivery partners can greatly simplify operations. Architectural choices related to security and compliance discussed later in this Guidebook will influence your choice of service delivery partners.

The application technology platform will have a major impact on operations. The choice of server operating system, storage type, application server, database platform, and coding language don’t just affect developers; they impact operations, too. Similarly, use of cloud technologies such as OpenStack, CloudStack, Microsoft Azure, Google Cloud Platform, and Amazon Web Services (AWS) will greatly impact SaaS operations.

Most public cloud providers offer proprietary services, features, and application programming interfaces (APIs). While these can simplify development, they can also lead to cloud vendor lock-in. Service delivery providers differ by their geographic reach and availability. It is often better to use services that are similar across multiple platforms. This allows SaaS providers to select the best service delivery provider in each geographic area.
3. Secure

Securing SaaS services is a critical and ongoing endeavor. Legacy client/server systems often restrict access to the software behind the corporate firewall or through Virtual Private Networks (VPNs). In contrast, modern systems offer worldwide access, providing many more points of attack. Therefore, SaaS and cloud-based systems lead to increased security challenges due to their wider security perimeter. As with systems architecture, securing SaaS operations requires a collaborative effort between development and operations teams as well as with service delivery partners.

Security design involves preventing breaches as much as possible and responding to breaches when, and if, they occur. Guiding design principles include secure by design, defense in depth, principle of least privilege, and separation of duties. The use of intrusion detection systems and intrusion prevention systems (IDS/IPS) at both network and application layers is also important.

Categories of SaaS Security Attacks

There are many different methods of targeting software or operational weaknesses. SaaS delivery teams must be constantly vigilant to prevent the compromise of their systems.

- **Software Attacks**
  - SaaS Software Deficiencies
  - System Software Vulnerabilities

- **Physical Access Exploits**
  - USB Malware
  - USB Data Theft

- **Personnel Exploits**
  - Social Engineering
  - Malicious Employees/Sys Admins

- **Trust Exploits**
  - Attacks via APIs
  - Untrustworthy Trusted Access

- **Advanced Persistent Threats**
  - Spyware/Malware “Laying in Wait”
  - Trojan Horses

- **Site Attacks**
  - Distributed Denial of Service
  - Defacing/Taking Control of Site

**Software Attacks**

Flaws in SaaS design may allow systems to be breached through browser, mobile, or API inputs. Common exploit techniques involve tricking the system into executing with high security privileges of the software. Examples of common exploits include SQL injection\(^6\) and cross-site scripting.\(^7\)

Vulnerabilities in the operating system and other third-party software can also enable malicious actors to gain access to portions of SaaS data (e.g., Heartbleed) or take control of the SaaS system (e.g., Shellshock). To minimize risk, delivery teams should follow a well-defined plan for patching and maintaining systems. Updates should be deployed and tested in a pre-production environment prior to production release as part of a comprehensive release management plan.

**Physical Access Exploits**

Physical access to servers, storage arrays, or any system that has access to secure data can lead to security breaches. It creates the opportunity to introduce viruses into the system (e.g., Stuxnet - Iran) or to remove sensitive data from the system (e.g., Snowden - CIA; Bradley Manning - U.S Army).

A particular risk with SaaS is the pervasive use of APIs. If not secured, attackers can use them to exploit the system. Developers should assume all access points to the system can be gateways for the introduction of malware (for example, the Target credit card database was breached through HVAC software).

Malware can be introduced through computers authorized to access the SaaS system. The malware then accesses the system through those infected computers with the credentials of authorized users.

To reduce risk, operations facilities should conform to industry standards such as SSAE16 SOC-1 Type II and ISO 27001. In addition, security best practices include multi-level biometric authentication for staff, continuous close circuit television (CCTV) monitoring and recording, detailed personnel auditing records, as well as 24 x 7 x 365 onsite security personnel.

---


Personnel Exploits

Unauthorized users can access systems by using social engineering to determine the security credentials of legitimate users. The use of identical passwords on multiple systems can result in unauthorized access propagating to many parts of your computing environment. Role Based Access Control should be followed to preclude unnecessary users. The use of strong password complexity and enforcing expiration policies are also important countermeasures to prevent breaches.

Trust Exploits

Malware can exploit both software and users with high trust levels. Multiple software systems can share the same security and firewalls. Gaining access to one system exposes data from a second, more secure system. To prevent cascading compromises, system access and administration should be isolated as much as possible. Ports and protocols should be locked down, and access should only be granted for specific needs.

Advanced persistent threats

Advanced Persistent Threats (APT) are long-term exploits that use malware to compromise critical systems. The attack lays dormant until called into action. This type of threat can slowly and surreptitiously transmit sensitive data to unauthorized external systems, leaving almost no traces of its actions. Ongoing systems security auditing along with data loss prevention tools can mitigate this kind of threat.

Site attacks

Instead of gaining access to SaaS systems and data, bad actors may act to prevent the SaaS system from functioning through distributed denial of service (DDoS) attacks (as we saw with the recent Sony data breach). This typically involves flooding inbound web servers from many endpoints, thereby overwhelming systems and preventing legitimate traffic. Several companies offer DDoS protection and mitigation services to limit the impact of this kind of threat.

Malware

Malware authors fall into any of these categories: stealing data, altering data, installing malware, taking control of the system or transactions on the systems and taking down SaaS operations.

The most publicized security breaches are the theft of credit card data and user names and passwords. Malware authors use the information to break into individual accounts to initiate authorized activity on the user’s behalf, presumably with the intent of financial gain or exploitation of private information (for example, Apple celebrity photo leak).
SaaS Operational Countermeasures

SaaS operations require multiple offensive and defensive strategies to protect against security breaches.

- Authentication
- Access Limits
- Firewalls
- Intrusion Detection
- Access Analysis

Access Controls

Limiting system access is fundamental to thwarting attacks. Strong password complexity requirements and multifactor authentication are also important deterrents. Access to systems and data should be kept to a minimum of system administration staff. Automated monitoring tools can help decrease the likelihood of security breaches.

Limiting the attack face (the number of external points of security vulnerability) of the SaaS system is important to reducing the threat level to the SaaS operation. Private network connectivity can greatly reduce the chances of unauthorized access.

Private Connectivity

Data that traverses public networks is vulnerable to attack. By contrast, private connectivity uses dedicated links. In addition to enhanced security, private connectivity allows for QoS management that is not possible over best-effort public links.

Private connectivity offers better performance for latency-sensitive and high-bandwidth traffic. It can also be a cost-effective means of linking distributed operations. Equinix offers private connectivity to over 1,000 networks and more than 450 cloud service providers across their global platform of International Business Exchange (IBX) data centers.
Security Partnerships

The defense against threats is a never-ending process. New exploits are constantly being introduced. Few SaaS companies have the expertise or breadth of experience of external resources required to harden their system against all potential threats. SaaS delivery teams should construct security processes with the assistance of external security experts. Additionally, it’s impractical for the same group to build the security mechanisms and test them thoroughly. Independent external providers should be used to audit and test SaaS security.
4. Comply

Compliance with regulatory standards is an important and often challenging part of SaaS service delivery. The task is compounded by the fact that regulations vary by location, industry, and whether a firm is public or private. Compliance with relevant standards is not optional and must be taken very seriously, since inability to meet compliance requirements can threaten the viability of a SaaS provider.

Common compliance focus areas include:

- Monitoring and mitigation of security threats
- Securing employee endpoint devices
- Ensuring data privacy and sovereignty
- Safeguarding personally identifiable information
- Definition and documentation of financial key controls
- Recordkeeping and retention requirements
- Data protection, including backup and/or replication
- Disaster recovery and business continuity planning
- Periodic third-party auditing
Compliance authorities

Many different authorities have compliance requirements. In some cases these may overlap:

<table>
<thead>
<tr>
<th>Compliance Authority</th>
<th>Compliance Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers SLAs</td>
<td>Security</td>
</tr>
<tr>
<td>Governmental Compliance</td>
<td>Privacy</td>
</tr>
<tr>
<td>Industry-Specific Compliance</td>
<td>Data Governance</td>
</tr>
<tr>
<td>Standards Bodies Compliance</td>
<td>Procedures</td>
</tr>
<tr>
<td></td>
<td>Availability</td>
</tr>
<tr>
<td></td>
<td>Disclosures</td>
</tr>
</tbody>
</table>

Customer SLAs

Effective SaaS systems must meet their customers’ functional requirements and offer quality assurances as specified in their SLA. The SLA will also specify the remedies for failure to meet these guarantees.

The SLA should clearly define the customer experience and the overall service quality from the SaaS provider. The SLA terms may include guarantees for uptime, performance levels, response times, disaster recovery assurances, privacy, security, and in some cases, specific business results. Monitoring adherence to these benchmarks is a fundamental part of the SaaS service delivery process. Many vendors provide a real-time system status (e.g., Salesforce’s website trust.salesforce.com).

Governmental compliance

Compliance with governmental regulations can be complex for a SaaS vendor. Lack of compliance can cause a company to fail or expose its executives to prosecution and imprisonment. For example, public companies based in the United States must comply with the Sarbanes-Oxley Act of 2002. This act consists of a series of regulations with detailed focus on processing of information, roles of people involved in key processes (particularly in segregation of duties), and reporting requirements. Penalties for noncompliance can be severe.8

Besides lawsuits and negative publicity, a corporate officer who does not comply or submits an inaccurate certification is subject to a fine of up to $1 million and ten years in prison, even if done mistakenly. If a wrong certification was submitted purposely, the fine can be up to $5 million and twenty years in prison.

SaaS providers may face greater challenges compared to traditional software companies, since they often have operations in multiple countries and regions. Regulations can vary by region and administrative jurisdiction. For example, European Union (EU) regulators have increasingly strict regulations on the location of data used in their countries and the privacy protection of their citizens.

Data sovereignty and government control

Controlling where data resides is an important consideration for SaaS delivery. Many governments have strict regulations related to the transit and storage of their citizens’ personally identifiable information (PII). Countries like Germany, France, Australia, Saudi Arabia, and China all make it very difficult to use data center resources outside of their borders. They may also require data to be encrypted while at rest and/or in transit. SaaS companies wishing to attract users in these countries often must operate in multiple international data center locations.

Conversely, some governments may subpoena information held by the SaaS provider or monitor customer information without notice. For example, the USA PATRIOT Act of 2001 greatly extended the government’s authority to access and monitor online data. This has led many SaaS operators to encrypt most Internet traffic and restrict the transfer of data and computing resources outside of country borders. Many large SaaS providers such as Apple, Microsoft, Google, and Yahoo are aggressively hardening their data security through encryption and limiting their own ability to access and decrypt their customers’ data. Key management must guarantee that only the end user, not the SaaS provider, can decrypt data wherever possible. Encryption keys for high-security applications may need to be securely destroyed when access is no longer required. SaaS providers may also be required to delete user data within a specified period of time if requested by the end user.

Industry-specific compliance

Beyond the requirements already outlined, many industries and jurisdictions impose additional requirements. Examples include:

- Payment Card Industry Data Security Standard (PCI DSS) for safeguarding credit card information
- The United States Health Insurance Portability and Accountability Act of 1996 (HIPAA) for healthcare
- The Financial Industry Regulatory Authority (FINRA) for securities transactions sanctioned by the U.S. Securities and Exchange Commission

Failure to comply with these regulations can result in fines, civil actions, and even potentially losing authority to operate.

Standards bodies compliance

There are several standards bodies that have created compliance regulations and frameworks which are endorsed by industry and governmental organizations. For example, the International Standards Organization (ISO) has defined numerous standards including ISO 27001 which specifies security requirements for IT systems that operate globally. Similarly, the Statement on Standards for Attestation Engagements (SSAE) is a regulation from the American Standards Board (ASB) that includes multiple Service Organization Control (SOC) standards. SSAE-16 SOC-2 defines multiple Trust Service Principles covering security, availability, processing integrity, confidentiality, and privacy. These standards bodies become de facto creators of regulations with which SaaS vendors must comply.

SaaS providers who operate in different countries will often need to seek certifications from multiple standards bodies. Partnering with a hosting provider who already maintains these certifications can greatly simplify the process.
Overlapping compliance areas

Each country’s compliance authority is focused on specific regulatory viewpoints when evaluating SaaS solutions. This results in potential regulatory conflict between different countries over compliance rules. For example, the US tries to balance free speech against national security; the EU is concerned with personal privacy; and China requires control over information.

Privacy policies

Historically, software providers have dictated privacy rights of individuals in their “Terms of Service” (TOS) unless covered by a specific regulation generally related to medical or financial data. Conversely, compliance authorities such as HIPAA have extensive regulation in areas such as patient privacy. HIPAA compliance requires rigorous encryption of patient data, restriction of access, and the implementation of security processes in SaaS software.

Privacy regulations are expanding rapidly beyond just medical and financial data. The EU has codified its regulations in the “Data Protection Directive” (“Directive 95/46/EC”). Companies must give notice when their data is being collected and why. Data collection requires a user’s consent, the access of the data is restricted, and there must be means to correct inaccurate data. The International Safe Harbor Privacy Principles provide a streamlined process for U.S. companies to comply with the EU Data Protection Directive. US SaaS companies should take advantage of this simplified, less risky framework for compliance.

Besides Europe, most countries have regulations for the control of PII. Disclosure of this information can heighten the risk of fraud or the disclosure of private information that many jurisdictions prohibit by law.

SaaS companies should also consider participating in TRUSTe. Companies “self-certify” that they follow the TRUSTe standards to safely collect and use customer data. The TRUSTe certificate indicates that the SaaS company complies with its own privacy statement as well as the TRUSTe program requirements. This helps provide a framework for basic privacy assurance.

SaaS providers hold extensive customer data, which allows them to better understand and benchmark how their customers use their systems. It’s essential that SaaS vendors obtain their customers’ permission to use this data to institute procedures specifying how the data will be anonymized to prevent disclosure of confidential customer information.

Disclosure of data breaches

An increasing requirement of both government regulations and customer demands are for greater operational transparency. Regulators demand the disclosure of security breaches so that other companies can harden their defenses. SaaS customers also want to know about any security breaches that could impact their data.
5. Operate

SaaS delivery excellence requires coordination across all aspects of SaaS operations. To be successful, operations teams must align the Architect, Secure, and Comply phases of systems development with the resources, processes, and technologies required. Operations is continuous and ongoing, and requires relentless focus on quality and continuous improvement.

While operations teams vary among companies, there are common process and focus areas in most service delivery organizations. As discussed earlier, frameworks like ITIL, ITSM, and CMMI can provide service delivery guidance.

A vital step in service delivery planning is establishing alignment of goals between development and operations. Otherwise they may work at cross-purposes. For example, the development team might be focused on rapid introduction of features and expansion to new markets while the operations team is focused on maximizing availability and minimizing cost. Establishment of a DevOps team as a bridge between software development and service operations can help prevent misalignment and ensure effective service design and operations.

Goals and Functions of SaaS Operations

In addition to the security and compliance goals defined in prior sections, SaaS operations are responsible for delivery of key objectives including availability, capacity, and efficiency. Both availability and capacity planning were covered earlier in the Architect section of this guidebook. The fact they are included again here, illustrates how architecture and operations form a self-reinforcing feedback loop. Service delivery planners create the initial systems design. Once the system is implemented, systems administrators measure its actual performance. The degree to which performance matches estimates is assessed, and the model is tuned for future designs.

Availability Management

Customers expect the services they pay for to be available. However, this comes at a cost. Increased availability requires increased investments in systems, tools, and personnel. Most SaaS companies will provide a standard 99.5% availability guarantee, because it allows planned downtime for systems maintenance. If customers expect continuous availability, though,
this additional downtime can come at a price. For example, if an hour of downtime results in $25,000 of lost revenues, the difference between 99.5% and 99.999% availability translates to more than $1 million in lost business. SaaS providers should analyze these economic trade-offs and establish uptime goals that maximize profitability.

<table>
<thead>
<tr>
<th>SLA%</th>
<th>Calculated Daily Outage in Minutes</th>
<th>Calculated Monthly Outage in Minutes</th>
<th>Calculated Monthly Outage in Hours</th>
<th>Calculated Annually Outage in Hours</th>
<th>Calculated Annually Outage in Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.999%</td>
<td>0.0144</td>
<td>0.432</td>
<td>0.0072</td>
<td>0.0864</td>
<td>0.0036</td>
</tr>
<tr>
<td>99.99%</td>
<td>0.144</td>
<td>4.32</td>
<td>0.072</td>
<td>0.864</td>
<td>0.036</td>
</tr>
<tr>
<td>99.95%</td>
<td>0.72</td>
<td>21.6</td>
<td>0.36</td>
<td>4.32</td>
<td>0.18</td>
</tr>
<tr>
<td>99.90%</td>
<td>1.44</td>
<td>43.2</td>
<td>0.72</td>
<td>8.64</td>
<td>0.36</td>
</tr>
<tr>
<td>99.50%</td>
<td>7.2</td>
<td>216</td>
<td>3.6</td>
<td>43.2</td>
<td>1.8</td>
</tr>
</tbody>
</table>

This is a table that illustrates uptime percentages.

Equinix data centers have an availability SLA of five nines (99.999%). This is much higher than most self-managed data centers achieve and can result in reduced downtime and increased profitability.

Because it can occur at any time, unplanned downtime is even more impactful to customer satisfaction and must be minimized. SaaS applications should be continuously monitored for any anomalies or disruptions. Dedicated Network Operations Centers and/or Service Operations Centers (SOCs) should be set up to monitor application performance in real-time. Incident response teams should be established with clear operational procedures based on incident type and severity. The primary goal of these dedicated operations centers is to minimize the Mean Time To Restore service. To this end, the establishment and use of service operations knowledge bases and systems “runbooks” is vital. Important components, interfaces, and code modules should be built and monitored to reduce time for troubleshooting and diagnosis.

Capacity Management

Management of capacity is a balancing act. Perpetually unused capacity is wasteful and costly, while insufficient capacity can result in application crashes or poor application performance. To accurately gauge system capacity, SaaS providers should instrument and monitor each operational component including servers, storage, networks, databases, web servers, application servers, and related systems. They should also configure and run system load and stress testing. These tests should generate different traffic loads and provide measures of user experience from all locations that serve a significant number of users.

Large providers may take this a step farther by also simulating failure of system components. Netflix created their Chaos Monkey service to randomly cause component failures. This lets them better understand system failure modes and to improve the robustness of their application architecture. For even moderately complex systems, this is a great way to understand and mitigate against large-scale outages.

Trend monitoring and forecasting is another key aspect of capacity management. By instrumenting and collecting consumption data on system resources, operations analysts can understand how the various application components behave and interact. For example, operations analysts can use temporal utilization data to develop metrics indicating how storage increases with transaction volume. This, in turn, can allow them to provision sufficient additional capacity in advance of anticipated load spikes from things like season peaks or special events and promotions.

**Operational Efficiency**

To maximize profitability, SaaS companies need to continually look for ways to drive down their operational costs. Common approaches include elimination of duplication and waste, designing for multi-tenancy, exploiting economies of scale, and assessing build-versus-buy trade-offs.

As discussed in the Architect section, system operations should be automated as much as possible. The advantages of automation include self-documenting process flow, lower operating costs, greater scalability, and higher reliability due to fewer human errors.

Multi-tenancy enables more efficient resource utilization. Virtualization technologies have proven this over more than a decade. As Moore's Law predicted, performance has continued to increase dramatically, and multi-tenancy enables service providers to take advantage of these performance increases by sharing resources across more than one user and workload.

Similarly, SaaS systems operations can take advantage of economies of scale by sharing expensive or highly-specialized resources. Some examples include networking devices such as core switches, load balancers, and firewalls; enterprise storage arrays; and blade servers.

Finally, as previously mentioned in the “Considerations for Delivery Partnerships” portion of the Architect section, delivery teams should periodically assess build-versus-buy and partnership options. If, for instance, a managed service provider can offer storage capacity at a better price and equivalent or better service level, it might make sense to shift that portion of the operation to them. Or perhaps backing up your data to a large public cloud provider would be less expensive than on-premises data backups. This is similar to the IT outsourcing trend that enterprises have been following for the past 25+ years. The important thing is to assess all dependencies and implications. A few side-effects to consider include the impact of contracted services on your own SLAs, liability for service non-conformance, and what would happen if the provider went out of business.
A Roadmap for Operational Excellence

As networking pioneer Bob Metcalf was purportedly fond of saying, “reliability is never having to say you’re sorry.”

This quote serves as a good guiding principle for service operations. Operations teams should establish and follow a delivery framework. All administrative, support, and development staff should be involved in its development and implementation. And it should be continuously reviewed, assessed, and optimized.

- Build
  - Build SaaS Software
  - Select Partners
  - Build the SaaS Environment
  - Test the System under Load
  - Create, Test and Document Operational Processes
  - Leverage Industry Best Practices
  - Partner For Security and Platform

- Automate
  - Automate Everything
  - Scale Up & Down Automatically
  - Question Every Manual Process

- Operate
  - Onboarding
  - Scaling/Bursting
  - Resilience
  - Governance/Compliance

- Measure
  - Monitor
  - Analyze
  - Benchmark
  - Audit

- Optimize
  - Identify Problem Areas
  - Use Analytics for Optimization
  - Optimize Technology Being Used
  - Leverage Declining Costs

6. Summary

Building a world-class SaaS service delivery operation is challenging. SaaS, as a business model, is relatively new and is still evolving.

SaaS companies face operational challenges not typically encountered by their on-premises application predecessors. Services must perform to many customers’ requirements and expectations while remaining secure and conforming to ever-evolving regulatory requirements. And they must be architected for multi-tenancy. To be competitive and profitable, service delivery costs must be relentlessly optimized. These challenges are compounded for global SaaS providers who must operate within different jurisdictions and sometimes conflicting sets of laws.

Designing and operating infrastructure for a worldwide customer base is substantially more challenging than serving a single location. Architecting and deploying distributed operations can be much more complex. Best practices for providing optimal user experience include building a high-performance, global network backbone and distributing data to multiple locations, close to end users. SaaS providers should design their delivery infrastructure to take advantage of best-in-class services in each location. They should work with strategic delivery partners who can help reduce complexity and increase agility.

SaaS providers that want to leverage a distributed network architecture must build on a solid foundation. Equinix delivers world-class facilities, operational excellence and unmatched global presence to give SaaS providers a competitive advantage in the market.

Always strive to differentiate the SaaS customer experience and offer superior quality of service to ensure high renewal rates and control costs.

While the challenges are substantial, SaaS development and operations teams that work together with their service delivery providers and partners can create a world-class SaaS platform that will help drive your company’s and your customers’ success.
## Service Delivery Checklist

<table>
<thead>
<tr>
<th>Question</th>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Have you established customer SLAs that meet or surpass industry standards?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you achieved certifications for all required regulatory standards (such as PCI DSS, SSAE-16 SOC, and ISO/IEC 27001)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you applied for or been granted EU Safe Harbor status?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you have a privacy policy in place that meets both customer and regulatory demands such as the EU Data Protection Directive?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you have a data governance plan in place that you validate at least annually for compliance?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you completed security audit by accredited experts?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you have a plan to detect and respond to security breaches?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you perform an annual review of hosting expenses to determine ways of further reducing costs?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is your SaaS cost of service (hosting, administration, licensing and support) at or below the industry best practice of 18% of revenue (as achieved by leading companies such as Workday, Salesforce or NetSuite)?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Question</td>
<td>☑️ YES</td>
<td>☑️ NO</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>Do you periodically stress test your SaaS infrastructure to ensure it can handle peak loads and deliver acceptable performance to all users?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you created a customer accessible uptime reporting system similar to trust.salesforce.com/trust/status?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you instrument, monitor, and report on all critical components of your service delivery infrastructure?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you have a process for forecasting and responding to future demand, based on seasonal fluctuations and anticipated growth?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you use availability zones to ensure an outage in one data center will not cause a loss of customer access or data?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you created and tested a disaster recovery plan?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Glossary of SaaS Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Persistent Threats</td>
<td>Malware that becomes embedded and remains undetected in a SaaS system for a lengthy period of time during which it monitors and ultimately compromise the system by sending information back to the hackers or ends up ultimately controlling the system.</td>
</tr>
<tr>
<td>Application Isolation</td>
<td>The degree to which one application shares resources with other tenants. At the extreme, all computing resources can remain isolated in a different facility to minimize the security risk from other applications and tenants. Multi-tenant SaaS applications, which share processes and databases, must take additional precautions to ensure that each tenant’s data and processes are isolated from other tenants and applications.</td>
</tr>
<tr>
<td>Cloud Bursting</td>
<td>The capability of a SaaS operation to move a portion of a workload from a private cloud to a public cloud to handle a spike in load.</td>
</tr>
<tr>
<td>Cloud Computing Platform</td>
<td>A set of services and APIs for providing cloud-hosting services such as Amazon AWS, Microsoft Azure, OpenStack and Google Cloud Platform. Moving SaaS applications from one cloud-computing platform to another is often non-trivial. Relying on the APIs and services of a specific platform may result in “lock in.”</td>
</tr>
<tr>
<td>IaaS</td>
<td>Infrastructure as a Service refers to a combination of hosting, hardware, provisioning and basic services needed to run a SaaS or cloud application, and delivered on a pay-as-you-go basis.</td>
</tr>
<tr>
<td>Managed Services Provider (MSP)</td>
<td>Usually a hosting provider that offers a high level of services, application management, monitoring, reporting, billing, and call center support.</td>
</tr>
<tr>
<td>Multi-Tenancy</td>
<td>A software architecture where a single instance of the software runs on a server, serving multiple client organizations referred to as tenants. Multi-tenancy is contrasted with a multi-instance architecture where separate software instances or hardware systems are set up for different clients.</td>
</tr>
<tr>
<td>On-Premises</td>
<td>The traditional method of installing and customizing software on the customer’s own computers that reside inside their own data center.</td>
</tr>
</tbody>
</table>
**Platform as a Service (PaaS)**

Development platforms, where the development tool itself is hosted in the cloud. PaaS provides high-level services which reduce the developer requirements to scale and operate the SaaS operation.

**Private Cloud**

Private cloud employs cloud-computing principles within a customer's own internal networks. The term refers to the same virtualization and highly flexible and scalable methods used in large Internet-based enterprise data centers.

**Software as a Service (SaaS)**

SaaS refers to multi-tenant software delivered over the Internet. Customers consume the product as a subscription service, on a pay-as-you-go basis.

**Service Level Agreement (SLA)**

The contractual terms of service associated with a SaaS provider's offerings.

**U.S.-EU Safe Harbor**

A streamlined process for U.S. companies to request certification to comply with the EU Directive 95/46/EC on the protection of personal data.

**Zero Day Attack**

A zero day attack exploits a previously unknown weakness in the software before an operations team is able to patch the newly discovered vulnerability.
DISCLAIMER
This document has been prepared based on research, information and data as available at the date of publication without any independent verification. Montclare does not guarantee or warrant the accuracy, reliability, completeness or currency of the data, facts, opinions, information and all further content (collectively referred to herein as “content”) provided herein of any nature nor its commercial or other usefulness regarding any particular purpose, even if you have provided to us your desire to use the contents of this document for any specific purpose. You agree to conduct all reasonable due diligence prior to relying upon any content provided herein.
The technology sector, particularly where Software as a Service is involved, is a rapidly changing, dynamic sector of commerce. Some relevant information may not be available to us as we conduct our research, due to various factors, which may include trade secrets we are not privy to, patent applications that are as of yet unpublished and potentially disruptive technologies as yet to emerge in the marketplace, as well as other information that is outside of the scope of our current research or our ability to research and analyze. You have the sole responsibility for assessing the relevance and accuracy of the content of this publication. You agree to hold Montclare harmless from any loss, damage, cost or expense incurred or arising from your reliance upon the content of this document. You agree to independently investigate and verify any and all content, statements, concepts, suggestions, advice, data, information and claims presented in this document prior to relying on any such content for the purpose of making business-related decisions. This document should be considered a “starting point” to help you start your individual research and due-diligence related investigations. Specific products or services may be mentioned in this document but such references are not intended to be an endorsement or recommendation of any product or service, unless specifically stated otherwise. You agree to refrain from using any specific product or service unless and until you have independently verified that such usage is appropriate for your business. No specific results are promised from your usage of this document and the content herein.
There is a natural element of uncertainty and risk in relying upon the content contained herein, and you agree to accept the sole responsibility of such potential uncertainty and risk and waive any and all claims related to any losses, damages, harm, and costs related thereto such usage.
The document, and the content herein, is provided “as is” and Montclare disclaims any and all representations and warranties, whether express or implied, including, but not limited to the implied warranties of title, merchantability, fitness for any particular purpose or non-infringement.
No content contained herein should be deemed “investment advice.”

COPYRIGHT
This publication is protected by the United States Copyright Act and related international conventions and may not be reproduced in any form, in whole or in part, without prior written permission by Montclare. © Montclare, Inc.