CHAPTER 2

Service Level Management

Service Level Management (SLM) is a key for delivering the services that are necessary to remain competitive in the Internet environment. Service quality must remain stable and acceptable even when there are substantial changes in service volumes, customer activities, and the supporting infrastructures.

Superior service quality also becomes a competitive differentiator because it reduces customer churn and brings in new customers who are willing to pay the premiums for guaranteed service quality. Customer churn is an insidious problem for almost every service provider.

The competitive market increases customer acquisition costs because continuous marketing and promotions are necessary just to replace the eroding customer base. Higher customer acquisition costs must be dealt with by either raising prices (a difficult move in a highly competitive market) or by taking longer to amortize the acquisition costs before profitability for each customer is achieved. Improving customer retention therefore dramatically increases profits.

This chapter covers the basics of SLM and lays part of the groundwork for the rest of the book:

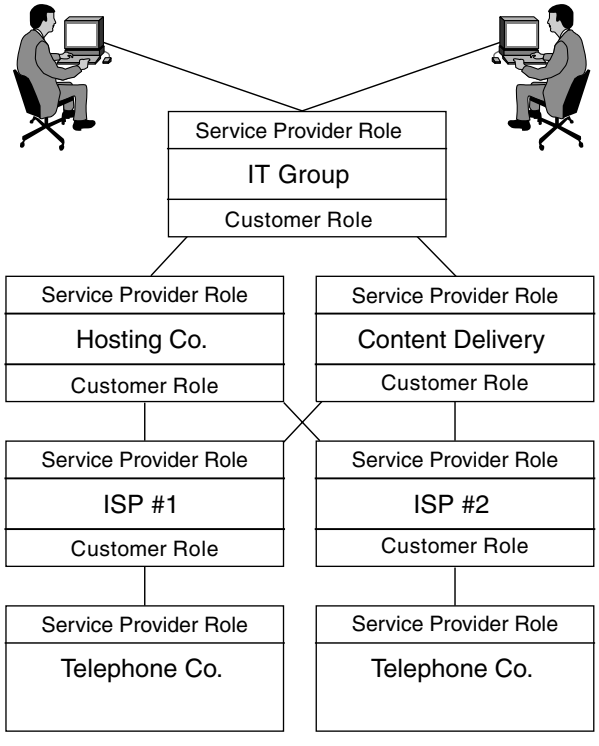
- An overview of SLM
- An introduction to technical metrics
- Detailed discussions of measurement granularity and measurement validation
- Business process metrics
- Service Level Agreements (SLAs)

Note that the chapter ends with a summary discussion in the context of building an SLA. Use of metrics in combination with the SLA's service level objectives to control performance is discussed in Chapter 6, "Real-Time Operations," and Chapter 7, "Policy-Based Management."

Overview of Service Level Management

Often, one group's service provider is another group's customer. It is critical to understand that service delivery is often, in fact, a chain of such relationships. As Figure 2-1 shows, some entities, such as an IT group, can play different roles in the service delivery process. As shown in the figure, a hosting company can be a customer of multiple service providers while in turn acting as a service provider. An IT group may be a customer of several service providers offering basic Internet connectivity, application hosting, content delivery, or other services. Customers may use multiple providers of the same service to increase their availability and to protect against dependence on a single provider. Customers will also use specialized service providers to fulfill particular needs.

Figure 2-1 *Roles of Customers and Service Providers*



The Internal Role of the IT Group

An IT group serves the entire organization by aggregating demands of individual business units and using them as leverage to reduce overall costs from service providers.

Today, such IT groups are making the necessary adjustments as managed services become a mandatory requirement. IT managers are constantly reassessing the business and strategic

trade-offs of developing internal competence and expertise as opposed to outsourcing more of the traditional IT work to external providers. The goal is to save money, protect strategic assets, and maintain the necessary flexibility to meet new challenges.

The External Role of the IT Group

IT groups are increasingly being required to provide specific levels of service, and they are also more frequently involved in helping business units negotiate agreements with external service providers. Business units often choose to deal directly with service providers when they have specialized needs or when they determine that the IT group cannot offer services with competitive costs and benefits.

IT groups must therefore manage their own service levels as well as those of service providers, and they must track compliance with negotiated SLAs.

The Components of Service Level Management

The process of monitoring service quality, detecting potential or actual problems, taking actions necessary to maintain or restore the necessary service quality, and reporting on achieved service levels is the core of SLM. Effective SLM solutions must deliver acceptable service quality at an acceptable price.

Acceptable quality from a customer perspective means an ability to use the managed services effectively. For example, acceptable quality may mean that an external customer or business partner can transact the necessary business that will generate revenues, strengthen business partnerships, increase the Internet brand, or improve internal productivity. Specific ongoing measurements are carried out to determine acceptable service quality levels, and noncompliance is noted and reported.

Acceptable costs must also be considered, because over-provisioning and throwing money at service quality problems is not an acceptable strategy for either service providers or their customers (and in spite of the cost, it often doesn't solve the problem). Service management policies are applied to critical resources so that they are allocated to the appropriate services; inappropriate activities are curtailed. Service providers that manage resources effectively deliver superior service quality at competitive prices. Their customers, in turn, must also increase their online business effectiveness and strengthen their bottom-line results.

The Participants in a Service Level Agreement

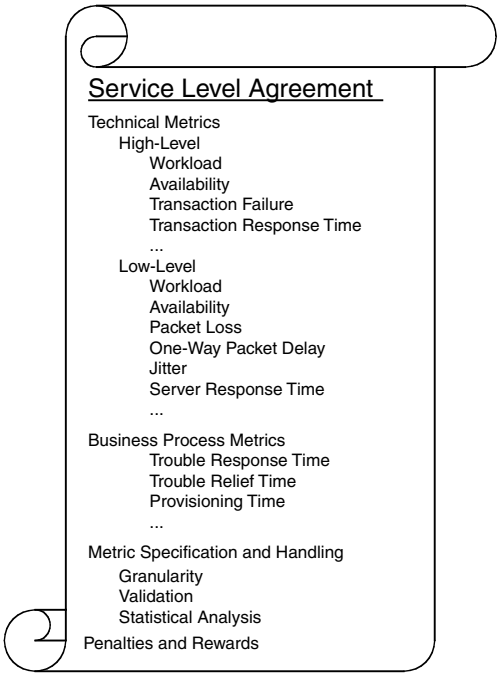
The SLA is the basic tool used to define acceptable quality and any relationships between quality and price. Because the SLA has value for both providers and customers, it's a wonder why it has taken so long for it to become important. In practice, many organizations

and providers find the process of negotiating an acceptable SLA to be a difficult task. As with many technical offerings, customers often experience difficulty in expressing what they need in technical terms that are both measurable and manageable; therefore, they have difficulty specifying their needs precisely and verifying that they are getting what they pay for. Service providers, on the other hand, appreciate clearly-specified requirements and want to take advantage of the opportunity to offer profitable premium services, but they also want to minimize the risks of public failure and avoid increasingly stringent financial penalties for noncompliance with the terms of the SLA.

Metrics Within a Service Level Agreement

Measurement is a key part of an SLA, and most SLAs have two different classes of metrics, as shown in Figure 2-2, which may be divided into technical metrics and business process metrics. *Technical metrics* include both high-level technical metrics, such as the success rate of an entire transaction as seen by an end user, and low-level technical metrics, such as the error rate of an underlying communications network. *Business process metrics* include measures of provider business practices, such as the speed with which they respond to problem reports.

Figure 2-2 *Contents of a Service Level Agreement*



Service providers may package the metrics into specific profiles that suit common customer requirements while simplifying the process of selecting and specifying the parameters. Service profiles help the service provider by simplifying their planning and resource allocation operations.

Introduction to Technical Metrics

Technical metrics are a core component of SLAs. They are used to quantify and to assess the key technical attributes of delivered services.

Examples of technical metrics are shown in Table 2-1. They are separated into the two basic groups: *high-level metrics*, which deal with attributes that are highly relevant to end users and are easily understood by them, and *low-level metrics*, which deal with attributes of the underlying technologies. Note that you should be very specific when defining these terms in an agreement. Although many of these terms are in common use, their definitions vary.

Table 2-1 *Examples of Technical Metrics*

Metric	Description
High-Level Technical Metrics	
Workload	Applied workload in terms understandable by the end user (such as end-user transactions/second)
Availability	Percentage of scheduled uptime that the system is perceived as available and functioning by the end user
Transaction Failure Rate	Percentage of initiated end-user transactions that fail to complete
Transaction Response Time	Measure of response-time characteristics of a user transaction
File Transfer Time	Measure of total transfer-time characteristics of a file transfer
Stream Quality	Measure of the user-perceived quality of a multimedia stream
Low-Level Technical Metrics	
Workload	Applied workload in terms relevant to underlying technologies (such as database transactions/second)
Availability	Percentage of scheduled uptime that the subsystem is available and functioning
Packet Loss	Measure of one-way packet loss characteristics between specified points
Latency	Measure of transit time characteristics between specified points
Jitter	Measure of the transit time variability characteristics between specified points
Server Response Time	Measure of response-time characteristics of particular server subsystems

Workload is an important characteristic of both high- and low-level metrics. It's not a measure of delivered quality; instead, it's a critical measure of the load applied to the system. For example, consider the workload of serving web pages. A text-only page might comprise only 10 K bytes, whereas a graphics page could comprise a few megabytes. If the requirement is to deliver a page in six seconds to the end user, massively different bandwidth and capacity will be necessary. Indeed, content may need to be altered for low-speed connections to meet the six-second download time.

NOTE In many situations, certain technical metrics aren't specified in the SLA. Instead, the supplier is asked to use *best effort*, which represents the classic Internet delivery strategy of "get it there somehow without concern for service quality." Today, best effort represents the commodity level for services. There are no special treatments for best-effort services. The only need is that there are sufficient resources to prevent best-effort services from *starving out*, which means having the connection time out because of long periods of inactivity.

Discussions of all of the examples in Table 2-1 follow, to illustrate the basic concepts of technical metrics. Additional descriptions of these metrics, and other technical metrics, appear in Chapters 4 and 8–10.

High-Level Technical Metrics

These metrics deal with workload and performance as seen and understood by the end user.

Workload

The *workload high-level technical metric* is the measure of applied load in end-user terms. It's unreasonable to expect a service provider to agree to service levels for an unspecified amount of workload; it's also unreasonable to expect that an end user will willingly substitute specification of obscurely-related low-level workload metrics instead of understandable high-level metrics. SLAs should therefore begin by specifying the high-level workload metrics, and service providers can then work with the customer's technical staff to derive low-level workload metrics from them.

For transaction systems, the workload metric is usually specified in terms of the end-user transaction mix and volumes, which typically vary according to time of day and other business cycles. For existing systems, these statistics can be obtained from logs; for new systems or situations (such as a proposed major advertising campaign designed to drive prospective customers to a web site), the organization's marketing group or their consultants should work to produce the most accurate, specific estimates possible. These workload estimates for new systems should be used for load testing as well as for SLAs.

Transaction workload metrics must include end-user tolerance for transaction response time delays. If response time delays are too long, external customers will abandon the transaction. In legacy systems where external customers did not interact directly with the server systems, abandonment was not a factor in workload testing. Call-center operators handled any delays by talking to the customers, shielding them from the problem, if necessary. On the Web, customers see the delays without any shielding, and they may decide at any point to abandon the transaction—with immediate impact on the server system's workload.

Another effect of the direct connection between customers and web-serving systems is that there's no buffer between those customers and the servers. In a call center, the workload is buffered by external queues. Incoming calls go through an automatic call distribution system; callers are placed on hold until an operator is available. In an order-entry center, the workload is buffered by the stack of documents on the entry clerk's desk. In contrast, the web workload has no external buffer; massive spikes in workload hit the servers instantly. These spikes in workload are called *flash load*, and they must be specified in the workload metric and considered during load testing. Load specification for the Web should therefore be in terms of arrival rate, not concurrent users, as was the case for call centers and order-entry centers.

File-serving, web-page, and streaming-media workload metrics are similar to transaction metrics, but simpler. They're usually specified in terms of the size and number of files that must be transferred in a given time interval. (For web pages, the types of the files are usually specified. Dynamically-generated files are clearly more resource-intensive than stored static files.) The serving system must have the bandwidth to serve the files, and it must also be able to handle the anticipated number of concurrent connections. There's a relationship between these two variables; given a certain arrival rate, higher end-to-end bandwidth results in fewer concurrent users.

Availability

Availability is the percentage of time that the system is perceived as available and functioning by the end user. It is a function of both the Mean Time Between Failures (MTBF) and the Mean Time To Repair (MTTR). Scheduled downtime might, in some organizations, be excluded from these calculations. In those organizations, a system can be declared 100 percent available even though it's down for an hour every night for system maintenance.

Availability is a binary measurement—the service is either available or it isn't. For the end user, and therefore for the high-level availability metric, the fact that particular underlying components of a service are unavailable is not a concern if that unavailability is concealed through redundant systems design.

Availability can be improved by increasing the MTBF or by decreasing the time spent on each failure, which is measured by the MTTR. Chapter 3, "Service Management

Architecture," introduces the concept of *triage*, which decreases MTTR through quick assignment of problems to the appropriate specialist organization.

Transaction Failure Rate

A transaction fails if, having successfully started, it does not successfully complete. (Failure to start is the result of an availability problem.) As is true for availability, systems design and redundancy may conceal some low-level failures from the end user and therefore exclude the failures from the high-level transaction failure rate metric.

Transaction Response Time

This metric represents the acceptable delay for completing a transaction, measured at the level of a business process.

It's important to measure both the total time to complete a transaction and the elapsed time per page of the transaction. That's because the end user's perception of transaction time, which will be used to compare your system with your competitors', is based on total transaction time, regardless of the number of pages involved, while the slowest page will influence end-user abandonment of a web transaction.

File Transfer Time

The file transfer time metric is closely associated with specified workload and is a measure of success. The file transfer workload metric describes the work that must be accomplished in a certain period; the file transfer time metric shows whether that workload was successfully handled. Lack of end-to-end bandwidth, an insufficient number of concurrent connections, or persistent transmission errors (requiring retransmission) will influence this measure.

Stream Quality

The quality of multimedia streams is difficult to measure. Although underlying low-level technical metrics, such as frame loss, can be obtained, their relationship to the quality as perceived by an end user is very complex.

Streaming is a real-time service in which the content continues flowing even with variations in the underlying data transmission rates and despite some underlying errors. A content consumer may see a small blemish on a graphic because a packet is lost in transit—equivalent to static on your car radio. There is no rewinding and playing it again, as there might be with interactive services. Thus, packet loss is handled by just continuing with the streaming rather than retransmitting lost packets.

Occasional packet loss can still be tolerated and sometimes may not even be noticed. If packet loss increases, quality will begin to degrade until it falls below a threshold and becomes unacceptable. Years of development have been focused on concealing these low-level errors from the multimedia consumer, and the major existing technologies from Microsoft, Real Networks, Apple, and others have different sensitivities to these errors.

Nevertheless, quality must be measured. The telephone companies years ago established the Mean Opinion Score (MOS), a measure of the quality of telephone voice transmission. There are also international standards for evaluation of audio and video quality as perceived by human end users; examples are the International Telecommunication Union's ITU-T P.800-series and P.900-series standards and the American National Standards Institute's T1.518 and T1.801 standards. Simpler methods are also in use, such as measuring the percentage of successful connection attempts to the streaming server, the effective bandwidth delivered over that connection, and the number of rebuffers during transmission.

Low-Level Technical Metrics

These metrics deal with workload and performance of the underlying technical subsystems, such as the transport infrastructure. Low-level technical metrics can be selected and defined by first understanding the high-level technical metrics and their implications for the performance requirements placed on underlying subsystems. For example, a clear understanding of required transaction response time and the associated transaction characteristics (the number of transits across the transport network, the size of each transit, and so on) can help set the objective for the low-level technical metric that measures network transit time (latency).

Workload and Availability

These low-level technical metrics are similar to those for the high-level discussion, but they're focused on performance characteristics of the underlying systems rather than on performance characteristics that are directly visible to end users. Their correlation with the high-level metrics depends on the particular system design and the degree of redundancy and substitution within that design.

Throughput, for example, is a low-level technical metric that measures the capacity of a particular service flow. Services with rich content or critical real-time requirements might need sufficient bandwidth to maintain acceptable service quality. Certain transactions, such as downloading a file or accessing a new web page, might also require a certain bandwidth for transferring rich content, such as complex graphics, within the specified transaction delay time.

Packet Loss

Packet loss has different effects on the end-user experience, depending on the service using the transport. The choice of a packet loss metric for a particular application must be carefully considered. For example, packet loss in file transfer forces retransmission unless the high-level transport contains embedded error correction codes. In contrast, moderate packet loss in streaming media may have no user-perceptible effect at all—unless bad luck results in the loss of a key frame.

The burst length must be included in packet loss metrics. Usually a uniform distribution of dropped packets over longer time intervals is implicitly assumed. For example, out of every 100 packets there could be two lost without violating an SLA calling for two percent packet loss. There may be a different perspective if you examine behavior over longer intervals, such as 1,000 packets. Up to 20 packets in a row could be lost without violating the SLA. However, losing 20 consecutive packets—creating a significant gap in data received—might drive quality levels to unacceptable values.

Latency

Latency is the time needed for transit across the network; it's critical for real-time services. Excessive latency quickly degrades the quality of web sites and of interactive sound and video.

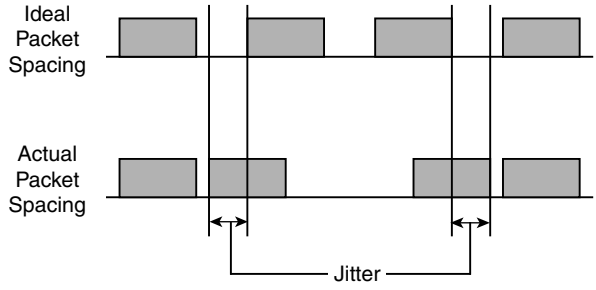
Routes in the Internet are usually asymmetric, with flows often taking different paths coming and going between any pair of locations. Thus, the delays in each direction are usually different. Fortunately, most Internet applications are primarily sensitive to round-trip delays, which are much simpler to measure than one-way delays. File transfer, web sites, and transactions all require a flow of acknowledgments in the opposite direction to data flow. If acknowledgments are delayed, transmission temporarily ceases. The round-trip latency therefore controls the effective bandwidth of the transmission.

Round-trip latency is much simpler to measure than one-way latency, because clock synchronization of separated locations is not necessary. That synchronization can be quite tricky if it is accomplished across the same network that's having its one-way delay measured. In that case, fluctuations in the metric that's being measured (one-way latency) can easily affect the stability of the measurement apparatus for one-way latency. An external reference, such as the satellite Global Positioning System (GPS) timers, is often used in such situations.

Jitter

Jitter is the deviation in the arrival rate of data from ideal, evenly-spaced arrival; see Figure 2-3. Some packets may be bunched more closely together (in terms of inter-packet delays) or spread farther apart after crossing the network infrastructure. Jitter is caused by the internal operation of network equipment, and it's unavoidable. Jitter is created whenever there are queues and buffering in a system. Extreme varieties of jitter are also created when there's rerouting of packets because of network congestion or failure.

Figure 2-3 *Jitter*



Interactive teleconferencing is an example of a service that is extremely sensitive to jitter; too much jitter can make the service completely useless. Therefore, a reduction in jitter, approaching zero, represents an increase in quality.

Buffering in the receiving device can be used to smooth out jitter; the jitter buffer is familiar to those of us who have a CD player in the car. Small bumps are smoothed out and the sound quality remains acceptable, but hitting a pothole usually causes more disturbance than the buffer can overcome. The dejitter buffer allows for latency that is typically one or two times that of the expected jitter; it's not a cure for all situations. The time spent in the dejitter buffers is an important contributor to total system latency.

Server Response Time

Similar to the high-level technical metric transaction response time, this measures the individual response time characteristics of underlying server systems. A common example is the response time of the database back-end systems to specific query types. Although not directly seen by end users, this is an important part of overall system performance.

Measurement Granularity

The SLA must describe the granularity of the measurements. There are three related parts to that granularity: the scope, the sampling frequency, and the aggregation interval.

Measurement Scope

The first consideration is the scope of the measurement, and availability metrics make an excellent example. Many providers define the availability of their services based on an overall average of availability across all access points. This is an approach that gives the service providers the most flexibility and cushion for meeting negotiated levels.

Consider if your company had 100 sites and a target of 99 percent availability based on an overall average. Ninety-nine of your sites could have complete availability (100 percent) while one could have zero. Having a site with an extended period of complete unavailability isn't usually acceptable, but the service provider has complied with the negotiated terms of the SLA.

If the availability level is specified on a per-site basis instead, the provider would have been found to be noncompliant and appropriate actions would follow in the form of penalties or lost customers. The same principle applies when measuring the availability of multiple sites, servers, or other units.

Availability has an additional scope dimension, in addition to breadth: the depth to which the end user can penetrate to the desired service. To use a telephone analogy, is dial tone sufficient, or must the end user be able to reach specific numbers? In other words, which transactions must be accessible for the system to be regarded as available?

Scope issues for performance metrics are similar to those for the availability metric. There may be different sets of metrics for different groups of transactions, different times of day, and different groups of end users. Some transactions may be unusually important to particular groups of end users at particular times and completely unimportant at other times.

Regardless of the scope selected for a given individual metric, it's important to realize that executive management will want these various metrics aggregated into a single measure of overall performance. Derivation of that aggregated metric must be addressed during measurement definition.

Measurement Sampling Frequency

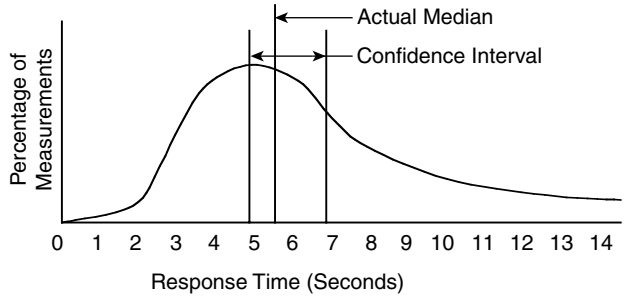
A shorter sampling frequency catches problems sooner at the expense of consuming additional network, server, and application resources. Longer intervals between measurements reduce the impacts while possibly missing important changes, or at least not detecting them as quickly as when a shorter interval is used. Customers and the service providers will need to negotiate the measurement interval because it affects the cost of the service to some extent.

Statisticians recommend that sampling be random because it avoids accidental synchronization with underlying processes and the resulting distortion of the metric. Random sampling also helps discover brief patterns of poor performance; consecutive bad results are more meaningful than individual, spaced-out difficulties.

Confidence interval calculations can be used to help determine the sampling frequency. Although it is impossible to perform an infinite number of measurements, it is possible to calculate a range of values that we're reasonably sure would contain the true summary values (median, average, and so on) if you could have performed an infinite number of measurements. For example, you might want to be able to say the following: "There's a 95

percent chance that the true median, if we could perform an infinite number of measurements, would be between five seconds and seven seconds." That is what the "95 Percent Confidence Interval" seeks to estimate, as shown in Figure 2-4. When you take more measurements, the confidence interval (two seconds in this example) usually becomes narrower. Therefore, confidence intervals can be used to help estimate how many measurements you'll need to obtain a given level of precision with statistical confidence.

Figure 2-4 *Confidence Interval for Internet Data*



There are simple techniques for calculating confidence intervals for "normal distributions" of data (the familiar bell-shaped curve). Unfortunately, as discussed in the subsequent section on statistical analysis, Internet distributions are so different from the "normal distribution" that these techniques cannot be used. Instead, the statistical simulation technique known as "bootstrapping" can be used for these calculations on Internet distributions.

In some cases, depending on the pattern of measurements, simple approximations for calculating confidence intervals may be used. Keynote Systems recommends the following calculation approximation for calculating the confidence interval for availability metrics. (This information is drawn from "Keynote Data Accuracy and Statistical Analysis for Performance Trending and Service Level Management," Keynote Systems Inc., San Mateo, California, 2002.) The formula is as follows:

- Omit data points that indicate measurement problems instead of availability problems.
- Calculate a preliminary estimate of the 95 percent confidence interval for average availability (avg) of a measurement sample with *n* valid data points:

  Preliminary 95 Percent Confidence Interval = avg ± (1.96 * square root [(avg * (1 – avg))/(*n* – 1)])

  For example, with a sample size *n* of 100, if 12 percent of the valid measurements are errors, the average availability is 88 percent. The confidence interval is calculated by the formula as (0.82, 0.94). This suggests that there's a 95 percent probability that the true average availability—if

we'd miraculously taken an infinite number of measurements—is between 82 and 94 percent. Notice that even with 100 measurements, this confidence interval leaves much room for uncertainty! To narrow that band, you need more valid measurements (a larger *n*, such as 1000 data points).

- Now you must decide if the preliminary calculations are reasonable. We suggest that the preliminary calculation should be accepted only if the upper limit is below 100 percent and the lower limit is above 0 percent. (The example just used gives an upper limit > 100% for *n* = 29 or fewer, so this rule suggests that the calculation is reasonable if *n* = 30 or greater.)

  Note that we're not saying that the confidence interval is too wide if the upper limit is above 100 percent (or if the average availability itself is 100 percent because no errors were detected); we're saying that you *don't know* what the confidence interval is. The reason is that the simplifying assumptions you used to construct the calculation break down if there are not enough data points.

For performance metrics, a simple solution to the problem of confidence intervals is to use geometric means and "geometric deviations" as measures of performance, which are described in the subsequent section in this chapter on statistical analysis.

Keynote Systems suggests, in the paper previously cited, that you can approximate the 95 Percent Confidence Interval for the geometric mean as follows, for a measurement sample with *n* valid (nonerror) data points:

Upper Limit = [geometric mean] * [(geometric deviation) $^{(1.96\ /\ (\text{square root of } [n-1]\ )\ )}$]

Lower Limit = [geometric mean] / [(geometric deviation) $^{(1.96\ /\ (\text{square root of } [n-1]\ )\ )}$]

This is similar to the use of the standard deviation with normally distributed data and can be used as a rough approximation of confidence intervals for performance measurements. Note that this ignores cyclic variations, such as by time of day or day of week; it is also somewhat distorted because even the logarithms of the original data are asymmetrically distributed, sometimes with a skew greater than 3. Nevertheless, the errors encountered using this recipe are much less than those that result from the usual use of mean and standard deviation.

Measurement Aggregation Interval

Selecting the time interval over which availability and performance are aggregated should also be considered. Generally, providers and customers agree upon time spans ranging from a week to a month. These are practical time intervals because they will tend to hide small fluctuations and irrelevant outlying measurements, but still enable reasonably prompt analysis and response. Longer intervals enable longer problem periods before the SLA is violated.

Table 2-2 shows this idea. If availability is measured on a small scale (hourly), high availability and requirements such as the *5-9's* or *99.999%* permit only 0.036 seconds of outage before there's a breach of the SLA. Providers must provision with adequate redundancy to meet this type of stringent requirement, and clearly they will pass on these costs to the customers that demand such high availability.

Table 2-2 *Measurement Aggregation Intervals for Availability*

Availability Percentage	Allowable Outage for Specified Aggregation Intervals			
	Hour	**Day**	**Week**	**4 Weeks**
98%	1.2 min	28.8 min	3.36 hr	13.4 hr
98.5%	0.9 min	21.6 min	2.52 hr	10 hr
99%	0.6 min	14.4 min	1.68 hr	6.7 hr
99.5%	0.3 min	7.2 min	50.4 min	3.36 hr
99.9%	3.6 sec	1.44 min	10 min	40 min
99.99%	0.36 sec	8.64 sec	1 min	4 min
99.999%	0.036 sec	0.864 sec	6 sec	24 sec

If a monthly (four-week) measurement interval is chosen, the 99.999 percent level indicates that an acceptable cumulative outage of 24 seconds per month is permitted while remaining in compliance. A 99.9 percent availability level permits up to 40 minutes of accumulated downtime for a service each month. Many providers are still trying to negotiate an SLA with availability levels ranging from 98 to 99.5 percent, or cumulative downtimes of 13.4 to 3.5 hours each month.

Note that these values assume 24 * 7 * 365 operations. For operations that do not require round-the-clock availability, or are not up during weekends, or have scheduled maintenance periods, the values will change. That said, they're pretty easy to compute.

The key is for service provider and service customer to set a common definition of the critical time interval. Because longer aggregation intervals permit longer periods during which metrics may be outside tolerance, many organizations must look more deeply at their aggregation definitions and look to their *tolerance for service interruption*. A 98 percent availability level may be adequate and also economically acceptable, but how would the business function if the 13.5 allotted hours of downtime per month occurred in a single outage? Could the business tolerate an interruption of that length without serious damage? If not, then another metric that limits the interruption must be incorporated. This could be expressed in a statement such as the following: "Monthly availability at all sites shall be 98 percent or higher, and no service outage shall exceed three minutes." In other words, a little arithmetic to evaluate scenarios for compliance goes a long way.

Measurement Validation and Statistical Analysis

The Internet and Web are extremely complex statistically. Invalid measurements and incorrect statistical analysis can easily lead to SLA violations and penalties, which may then fall apart when challenged by the service provider using a more appropriate analysis. Therefore, special care must be taken to discard invalid measurements and to use the appropriate statistical analysis methods.

Measurement Validation

Measurement problems, which are artifacts of the measurement process, are inevitable in any large-scale measurement system. The important issues are how quickly these errors are detected and tagged in the database, and the degree of engineering and business integrity that's applied to the process of error detection and tagging.

Measurement problems can be caused by instrument malfunction, such as a response timer that fails, and by synthetic transaction script failure, which leads to false transaction error reports. It can also be caused by abnormal congestion on a measurement tool's access link to the backbone network and by many other factors. These failures are of the measurement system, not of the system being measured. They therefore are best excluded from any SLA compliance metrics.

Detection and tagging of erroneous measurements may take time, sometimes up to a day or more, as the measurement team investigates the situation. Fortunately, SLA reports are not generally done in real time, and there's therefore an opportunity to detect and remove such measurements.

The same measurements will probably also be used for quick diagnosis, or triage, and that usage requires real-time reporting. There's therefore no chance to remove erroneous measurements before use, and the quick diagnosis techniques must themselves handle possible problems in the measurement system. Good, fast-acting artifact reduction techniques (discussed in Chapter 5, "Event Management") can eliminate a large number of misleading error messages and reduce the burden on the provider management system.

An emerging alternative is using a trusted, independent third-party to provide the monitoring and SLA compliance verification. The advantage in having an independent party providing information is both service providers and their customers could view this party as objective when they have disputes about delivered service quality.

Keynote Systems and Brix Networks are early movers into this market space. Keynote Systems provides a service, whereas Brix Networks provides an integrated set of software and hardware measurement devices to be installed and managed by the owner of the SLA. They both provide active, managed measurement devices placed at the service demarcation points between customers and providers or between different providers. (Other companies, such as Mercury Interactive and BMC, now offer similar services and software.)

The measurement devices, known as "agents" in the Keynote service and "verifier platforms" in the Brix service, carry out periodic service quality measurements. They collect information and reduce it to trends and baselines. There is also a real-time alerting component when the measurement device detects a noncompliant situation. Alerts are forwarded to the Keynote or BrixWorx operations center where they are logged and included in service level quality reports. As the Keynote system is a service, Keynote provides measurement device management and measurement validation.

Keynote and BrixWorx also offer integration with other management systems and support systems for reporting to customers, provisioning staff, and other back-office departments. Test suites for more detailed testing are also stored at the center and deployed to the measurement platforms as necessary.

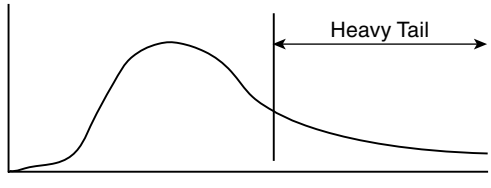
Trusted third parties may be the solution needed to reduce the problems when customer experience and provider data are not in close agreement.

Statistical Analysis

Most statistical behavior that you see in life is described by a "normal distribution," the typical "bell-shaped curve." This is an extremely convenient and well-understood data distribution, and much of our intuitive understanding of data is built on the assumption that the data we're examining fits the normal distribution. For a normal distribution, the arithmetic average is, indeed, the typical value of the data points, and a standard deviation calculated by the usual formula gives a good sense of the breadth of the distribution. (A small standard deviation implies a very tight grouping of data points around the average; a large standard deviation implies a loose grouping.) For a normal distribution, 67 percent of the measurements are within one standard deviation of the average, and 95 percent are within two standard deviations of the average.

Unfortunately, Web and Internet behavior do not conform to the normal distribution. As a result of intermixing long and short files, compressed video and acknowledgments, and retransmission timeouts, Internet performance has been shown to be heavy tailed with a right tail. (See Figure 2-5.) This means that a small but significant portion of the measurement data points will be much, much larger than the median.

Figure 2-5 *Heavy-Tailed Internet Data*



If you use just a few measurements to estimate an arithmetic average with a heavy-tailed distribution, the average will be very noisy. It's unpredictable whether one of the very large measurements will creep in and massively alter the whole average. Alternatively, you may be lulled into a false sense of security by not encountering such an outlying measurement (an *outlier*).

The situation for standard deviations is even worse because these are computed by squaring the distance from the arithmetic average. A single large measurement can therefore outweigh tens of thousands of typical measurements, creating a highly misleading standard deviation. It's mathematically computable, but worse than useless for business decisions. Use of arithmetic averages, standard deviations, and other statistical techniques that depend on an underlying normal distribution can therefore be quite misleading. They should certainly not be used for SLA contracts.

The geometric mean and the geometric standard deviation should be used for Internet measurements. Those measures are not only computationally manageable, they're also a good fit for an end-user's intuitive feeling for the "typical" measurement, psychologically. As an alternative, the median and eighty-fifth percentiles may be used, but they take more power to compute.

The geometric mean is the n^{th} root of the product of the n data points. The geometric deviation is the standard deviation of the data points in log space. The following algorithm should be used to avoid computational instabilities:

- Round up all zero values to a larger "threshold" value.
- Take the logarithm of the original measurements (any base).
- Perform any weighting you may want by replicating measurements.
- Take the arithmetic mean and the standard deviation of the logarithms of the original measurements.
- "Undo" the logarithm by exponentiating the results to the same base originally used.

Note that the geometric deviation is a factor; the geometric mean must be multiplied and divided by it to create the upper and lower deviations. Because of the use of logarithms, the upper and lower deviations are not symmetrical, as they are with a standard deviation in normal space. This is one of the prices you pay for the use of the geometric measures. Another disadvantage is that, as is also true for percentiles, you cannot simply add the geometric statistics for different measurements to get the geometric statistics for the sum of the measurements. For example, the geometric mean of (connection establishment time + file download time) is not the sum of the geometric means of the two components. Instead, each individual pair of data points must be individually combined before the computations are made.

These calculations of both the geometric mean and the geometric deviation, or the median and the eighty-fifth percentile, should be used for end-user response time specification. Using these statistics instead of conventional arithmetic averages or absolute maximums helps manage SLA violations effectively and avoids the expense of fixing violations that were caused by transient, unimportant problems.

Business Process Metrics

There have been numerous stories in industry publications that describe service provider difficulties in managing new technologies, digital subscriber line (DSL) services being a prime example. Customers were annoyed by the delays and operational interruptions. Many customers investigated alternative technologies with different service providers and subsequently left their original provider.

When customers defect, service providers suffer with lost business and revenues. Many startups in the DSL space, for example, could not deploy their services and generate revenue quickly enough and are out of business after exhausting their initial funding.

Many customers still view most of their providers as being behind the curve, sluggish, and unable to help them execute their business strategies fully. Typical complaints about interaction with providers often include the following:

- Difficulty in finding experts at the provider who actually understand the provider's own services
- Mistake-prone business processes for interacting with the provider
- Revenue impacts when scheduled services slip their delivery dates
- Voluminous, and often incomprehensible, bills and reports
- Bombardment from competitors offering equally incomprehensible services

Although such issues have made the service-provider marketplace somewhat turbulent, the good news is that the situation is improving because of two developments.

The first is the continuing build-out of the Internet core with optical transmission systems of tremendous capacity coupled to the widening deployment of broadband services for the last mile access links to the customer. When this capacity is fully in place, bandwidth services can be activated and deactivated without the delays associated with running new wiring and cable. As these high-capacity transmission systems become more widespread, it becomes a question of coordinating the activities of both customer and provider management systems for more effective and economical service delivery.

That introduces the second enabling factor: the development of standards, such as extensible markup language (XML) and Common Information Model (CIM), and other factors, are making the sharing of management information easier and simpler than it used to be.

Customers and service providers can use mechanisms such as XML to loosely couple their management systems. Neither party needs to expose internal information processes to the other, but they can exchange requests and information in real time to speed up and simplify their interactions.

Customers can allocate their spending more precisely by activating and deactivating services with finer control and thereby reducing their usage charges. They can also temporarily add capacity or services to accommodate sudden shifts in online business activities.

Providers have a competitive edge when they have the appropriate service management systems. They can meet customer needs quickly and use their own dynamic pricing strategies to generate additional revenues.

Business process metrics measure the quality of the interactions between customers and service providers as a way of including them in an SLA and thereby improving them. Some of these metrics may be incorporated in standard provider service profiles, while others may need to be negotiated explicitly.

Many customer organizations maintain relationships with multiple service providers to avoid depending on a single provider and to use the competition to extract the best prices and service quality they can negotiate.

Business process speed and accuracy will be even more important in the future as customer and provider management systems are integrated, and as services are activated and deactivated in real time. Service providers must be able to provision quickly, bill appropriately, and adjust services in a matter of a few seconds to a few minutes. Customers must also be able to understand their service mix and adjust their requests to the service provider to match changes in their business requirements. It is this environment that will begin to accelerate the use of business process metrics as part of the selection and continued evaluation of a set of service providers.

Table 2-3 lists two emerging categories of business process metrics. Problem management metrics measure the provider's responses to customer problems, whereas real-time service management metrics track the responses to customer requests for service modifications.

Table 2-3 *Business Process Metrics*

Metric	Description
Problem Management Metrics	
Trouble Response Time	Elapsed time between trouble notification by customer and first response by provider
Notification Time	Elapsed time between trouble detection by provider and first notification to the customer
Escalation Time	Elapsed time between first response by provider or notification to the customer and the first escalation to provider specialists
Trouble Relief Time	Elapsed time between first response by provider or notification to the customer and the furnishing of a workaround or fix for the problem that permits normal operation to resume
Trouble Resolution Time	Elapsed time between first response by provider or notification to the customer and the furnishing of a permanent fix for the problem
Real-Time Service Management Metrics	
Provisioning Time	The elapsed time to provision a new service
Activation / Deactivation Time	The elapsed time to activate or deactivate a provisioned service
Change Latency	The elapsed time to effect a parameter change across the entire system

The following sections describe the nuances of each metric in turn.

Problem Management Metrics

Service quality problems are inevitable, although, ideally, they are becoming more rare with time. A metric of primary importance is the *trouble response time* to a customer problem report or query. This metric can be used to measure both the first response to a customer call and the first response to automated notification from a customer's management system.

Notification time measures the interval between the provider detecting a service problem and reporting it to the customer. Agile customers will activate their own procedures to deal with the interruption and will want a quick notification time to minimize any disruptions.

Escalation time measures how quickly a problem is moved from the intake at the help desk to more highly qualified experts. Faster escalation times will usually carry a premium the customers will be willing to pay when critical services are involved. As is true for other problem management metrics, escalation time may depend on the severity of the problem and the priority assigned to the user's request.

Trouble relief time is that point at which the customer reporting the issue has a workaround in hand, or has overcome the service interruption. Relief is distinct from resolution: even if it's not known what caused the outage, the customer is back in business. However, the customer will want the provider to promptly identify the *root cause* of the outage and take corrective action to prevent it from happening again. That final stage is known as *resolution time*.

Real-Time Service Management Metrics

Customers and providers will both exploit real-time service management capabilities as their management systems begin to interact with each other. Customers will be able to fine tune their resource usage and control their costs while coping with the dynamic shifts that are so characteristic of online activities. Service providers will also have the advantage of maintaining control while allowing their customers to take over many of their management tasks and thereby reducing their staffing costs substantially.

Customers want to be able to change their service environment on their time schedule rather than waiting for the provider to do the job in the traditional way. This may involve, for example, activating services, such as videoconferencing, on a demand basis. At other times, customers may want to add capacity to handle temporary traffic surges, or they may want to change the priorities (and costs) of some of the services they use.

Provisioning time is the time needed to configure and prepare a new service for activation, including the allocation of resources and the explicit association of consumers with those resources for billing purposes. *Activation/deactivation time* is the time needed to activate or deactivate a provisioned service.

Change latency is an idea for a metric that arose from the experience of one of my colleagues. She works for a large multinational organization with approximately 1,200 global access devices. Some access points support a small number of dial-in users, while others accommodate larger buildings and campuses. Her organization wanted to change some access control policies and asked the service provider to update all the access devices. The problem occurred because the provider changed only portions of the devices in phases over two days rather than all at once, leading to a situation in which devices had inconsistent access control information. The result was disruptions to the business.

Service Level Agreements

The SLA has become an important concern for both providers and their customers as dependence on high-quality Internet services increases. The SLA is a negotiated agreement between service providers and their customers, and in the best of worlds, the SLA is explicit, complete, and easily understood. When done properly, the SLA serves the needs of both customers and service providers.

Organizations are constantly struggling to maintain or extend their competitive advantage with stable, highly available services for their customers and end users. As a result, they are increasingly dependent upon their service providers to deliver the consistent and predictable service levels on which their businesses depend. A well-crafted SLA provides substantial value for customers because of the following:

- They have an explicit agreement that defines the services that will be provided, the metrics to assess service provider performance, the measurements that are required, and the penalties for noncompliance.
- The clarity of the SLA removes much of the ambiguity in customer-service provider communication. The metrics, rather than arguments based on subjective opinions of whether the response time is acceptable, are the determinant for compliance.
- The SLA also helps customers manage their costs because they can allocate their spending on a differentiated scale with premiums for critical services and commodity pricing where best effort is sufficient.
- Customers have the confidence that they can successfully deploy the critical services that improve their internal operations (remote training and web-based internal services) or strengthen their ability to compete (web services and supply chains). Too many efforts have floundered due to unacceptable service quality after deployment.
- The SLA becomes more important as you move toward customer-managed service activation and resource management. The SLA will determine what the customer is allowed to do in real time in terms of changing priorities and service selections.

Service providers have been reluctant to negotiate SLAs because of their increased exposure to financial penalties and potentially adverse publicity if they fail to meet customer needs. In spite of their reluctance, they have been forced into adopting SLAs to keep their major customers. The evolution of SLAs has therefore been driven mainly by customer demands and fear of losing business.

Early SLAs focused primarily on availability because it was easier to measure and show compliance. Availability is also easier for a provider to supply by investing in the appropriate degree of redundancy so that failures do not have a significant impact on availability levels.

Performance metrics are beginning to be included in more SLAs because customers demand them. Providers have a more difficult time guaranteeing performance levels because of the dynamism of their shared infrastructures. Simply adding more bandwidth will not guarantee acceptable response time without significant traffic engineering, measurements, and continued analysis and adjustment. The difficulty of managing highly dynamic flows has many providers reluctant to accept the financial penalties that are part of most SLAs.

Nonetheless, the value of the SLA to providers is also recognized, and some of the significant factors are as follows:

- The clarity of the SLA serves the provider as it does the customer. Clearly defined metrics simplify the assignment of responsibility when service levels are questioned.
- The SLA offers service providers the capacity to differentiate their services and escape (somewhat) the struggles of competing in a commodity-based market. As providers create and deploy new services, they can charge on a value-pricing basis to increase their profit margins.
- High performance and availability are increasingly becoming competitive differentiators for service providers. Increasing customer dependence on Internet, content delivery, and hosting service providers gives an advantage to those providers that demonstrate their ability to deliver guaranteed service quality levels.

When constructing an SLA, customers must assess their desired mix of services and weigh their relative priorities. A useful first attempt is to match those needs against the providers' preconfigured service profiles. This will group services with common characteristics and requirements, and it will also help identify any special services that are not easily accommodated by the predefined categories. Requirements that do not fit a predefined class will require special considerations when negotiating an SLA.

After services have been grouped, their relative priorities within each category must be established. Customers can do this by selecting the appropriate service profile; for example, many service providers offer a variation on the platinum, gold, and silver profiles. Typically, platinum services are the most expensive and provide the highest quality; gold and silver are increasingly less expensive and provide relatively lower quality.

Even if prebuilt service profiles are used, the SLA negotiations must include discussions of how the SLA metrics are to be measured and how any penalties or rewards are to be calculated. Customers will continue to push for stronger financial penalties for noncompliance, and providers will give in to the pressure as slowly as they can in a highly competitive market.

Unfortunately, it's not uncommon for providers and customers to have ongoing disputes about the delivered services and their quality. Some of the roots of the problem are technical: customers and providers may have different measurement and monitoring capabilities and are therefore "comparing apples to oranges." Other problems are rooted in the terms of the SLA, where ambiguities lead to different interpretations. SLAs must therefore incorporate relevant measurement, artifact reduction, verification mechanisms, and appropriate statistical treatments to protect both parties as much as possible. Customers must play a role in the verification process because they still have the most to lose when serious service disruptions occur.

SLA penalties and rewards are a form of risk management on the part of the customer. However, they continue to be among the least well-developed elements of service offerings. More mature industries offer guarantees and incentives; the ability of the service provider to reduce and absorb some risk for its customers is a key competitive differentiator.

Still, customers bear the brunt of any disruptions caused by a provider. As one customer once said, "The problem is the punishment doesn't fit the crime; an hour-long outage costs us over $100,000, and my provider just gives me a 10 percent rebate on my next bill." Nevertheless, the correct role for penalties and rewards is to encourage good performance, not to compensate the customer for all losses. If loss compensation is needed, it's a job for risk insurance.

Rather, SLA penalties and rewards must focus on motivation. The penalties and rewards should be sufficient to inspire the performance the customer wants, and the goals should be set to ensure that the motivating quality of the SLA remains throughout the time period.

Impossible or trivial goals don't motivate, and capped penalties or goals stop motivating when the cap is reached. For example, if a provider must pay a penalty based on monthly performance, and the SLA is violated in the first three days of the month—so the maximum penalty must be paid—the provider won't be motivated to handle problems that appear during the remainder of the month. After all, that particular customer's ship has already sunk; maybe another customer's ship is still sinking and can be rescued without paying a maximum penalty!

Web performance goals that are set unrealistically high, with no reference to the Internet's background behavior, will cause the supplier to refuse the SLA or insist on minor penalties. A solution to this problem is to include in the SLA metrics a background measure of Internet performance or of competitors' performance, possibly from a public performance index or from specific measurements undertaken as a part of the SLA.

Sometimes, performance is so poor that a contract must be terminated. The SLA should discuss the conditions under which termination is an option, and it should also discuss who bears the costs for that termination. Again, the costs should be primarily designed to motivate the supplier to avoid terminations; it may not be possible to agree on an SLA in which all of the customer's termination costs are repaid.

Finally, customers may want to include security concerns in their SLA as part of a service profile through additional negotiation and specification. Security is notoriously difficult to measure, except in very large aggregates. Security metrics are more likely to take the form of response-time commitments in the event of a breach, either to roll out patches, shut down access, or detect an intrusion. The bulk of security discussions around service levels will be about policies, not measurement.

Summary

This chapter covers a lot of territory and sets the stage for the following chapter discussions that cover different aspects of actually managing services. Successful service management is predicated on delivering acceptable service quality at acceptable price points and within acceptable time frames. Correctly handled, it improves service quality, improves relationships with suppliers, and may even lower total costs.

The SLA is the basic tool used to define acceptable quality and any relationships between quality and price. It is a formal, negotiated contract between a service provider and a service user that defines the services to be provided, the service quality goals (often called service level indicators and service level objectives), and the actions to be taken if the service provider does not comply with the SLA terms.

Measurement is a key part of an SLA, and most SLAs have two different classes of metrics, technical and business process metrics. Technical metrics include both high-level technical metrics, such as the success rate of an entire transaction as seen by an end user, and low-level technical metrics, such as the error rate of an underlying communications network. Business process metrics include measures of provider business practices, such as the speed with which they respond to problem reports. Metrics should also include measures of the workload expected. Service providers may package the metrics into specific profiles that suit common customer requirements while simplifying the process of selecting and specifying the parameters.

In any case, a properly constructed SLA is based on metrics that are relevant to the end-user experience. Many of the low-level technical metrics, such as communications packet loss, have complex relationships to end-user experience; it's usually much better to use high-level technical metrics that directly measure end-user experience, such as web page download time and transaction time. The low-level technical metrics can then be derived from the high-level technical metrics and used to manage subordinate systems.

SLA metrics must be carefully defined in terms of scope, sampling frequency, and aggregation interval:

- Scope represents the breadth of measurement (for example, the number of test points from which availability is measured and the percentage of them that must be unavailable for the entire system to be marked as unavailable).

- Measurement sampling should be random, and the sampling frequency should be chosen to provide timely alerts when problems occur and to provide the appropriate confidence intervals for availability and performance measurement. Calculation of confidence intervals is unfortunately complex for Internet statistics, as the usual formulas, suited for normal distributions, cannot be used. Instead, statistical simulation through bootstrapping or the approximations discussed in the body of this chapter can provide estimates of the number of measurements needed to provide reasonable statistics.
- The aggregation interval is also important, as longer intervals, often chosen in SLAs, may allow long periods of sub-par performance. The tolerance for service interruption then becomes important and may need to be separately specified.

Measurements must also be validated and subjected to statistical treatment when used in SLAs, and the methods for that validation and treatment must be documented in the SLA to avoid dispute. Validation ensures that erroneous measurements are removed, insofar as is possible, before computation of the metrics used in the SLA. Statistical treatment ensures that outlying measurements do not create a misleading picture of the performance as perceived by end users, with the resulting waste of resources spent fixing what may be a minor issue. Arithmetic averages and standard deviations should not be used to handle Internet statistics.

Finally, the SLA should be written with penalty and reward clauses that are sufficient to inspire the performance the customer wants, and the goals should be set to ensure that the motivating quality of the SLA remains throughout the time period. Capped penalties or goals are examples of techniques that may motivate a supplier to abandon work on an account just because the cap has been reached—probably not the desired behavior.

The service level indicators and objectives described in the SLA are then used by the operations staff and by automated systems to manage the service levels, as described in Chapters 6 and 7.