COMMON DISTRIBUTED LOGGING

The distributed logging capability can be viewed as typical messaging applications where message producers generate log messages (e.g., informational, trace, error, or debug messages), and which may or may not be consumed by the interested message consumers over a period of time. The OGSA logging facility is an architecture model to separate the OGSA logging specific implementations to an intermediary or logging service.

These logging services or intermediaries should provide facilities for:

1. Decoupling. This helps to provide a clear separation of the roles of the log producers and log consumers. The producer has no previous knowledge of the message consumers, and how the message gets transformed.
2. Transformation and common representation. This facility provides plug-in transformation scripts to convert from one log format to another. Most notable among these transformation scripts is XSLT, which acts as an XML data transformation script. There is also a desirable approach to convert to a common data format based on a "common logging schema" representation suitable for canonical representation. This canonical schema can eliminate some transformation process and reduce the processing overheads.
3. Filtering and aggregation. Most of the logging may result in a huge amount of data and filtering of these data into certain buckets of desirable segments; this is a value-added feature. The OGSA logging service provides registration of such filtering criteria for each consumer, and aggregates the messages based on these criteria.
4. Configurable persistency. The durability of the logs is a major feature provided by the OGSA framework. We can enable this feature based on a per service and/or a per message basis (i.e., On Demand). For example, security logs and audits are kept intact for years to retrace some security vulnerability later on, should computer forensics become an issue.
5. Consumption patterns. Logging services should provide both synchronous (pull) and asynchronous (push) models of interaction of messages by the consumers. This service must provide out-of-band messaging facilities for critical messages and logs.

6. Secure logging. As we already know, many of these logs are critical, sensitive, and private; therefore, the need to store them and transport them in a secure fashion is an absolute requirement.

Figure 8.1 shows the OGSA logging service architecture and message flows. This facility utilizes the OGSI Notification framework for messaging semantics and includes consumer subscription for logs by providing filtering criteria and message delivery end points. The logs are delivered as messages based on specific notification topic changes and filtering criteria.

Figure 8.1. The OGSA logging service architecture model.