# *Symmetric Key Services Markup Language Requirements*

The OASIS Symmetric Key Services Markup Language (**SKSML**) is the proposed language/protocol that defines how a client on a network will request and receive services for symmetric encryption cryptographic keys from a server. This document establishes the requirements for SKSML, as well as the rationale for those requirements.

Clients may consist of computerized devices such as Personal Digital Assistants (PDA), telephones, laptop, desktop and server-class computers, applications such as office productivity, database, e-commerce, healthcare, financial or other applications, and/or devices such as routers, printers, disks, tape-drives, etc. Symmetric encryption cryptographic keys may consist of Triple Data Encryption Standard (3DES) or the Advanced Encryption Standard (AES). Please note that SKSML is not restricted to these examples; these are used, merely, as an illustration.

## Requirements

1. The language/protocol **must** be based on the eXtensible Markup Language (XML).
2. The language/protocol **must** use public-key cryptography for security.
3. The language/protocol **must NOT** rely on media or transport-level security.
4. The language/protocol **must** leverage existing standards where possible.

## Rationale

### I. The language/protocol must be based on the eXtensible Markup Language (XML).

While SKSML could be based on any language, XML is chosen for the following reasons:

a) It is the "lingua franca" of the 21st century internet. (OASIS lists 45 standards at http://www.oasis-open.org/specs/index.php, almost all of which are based on XML. The World Wide Web (W3C) consortium lists 103 Recommendations at http://www.w3.org/TR/#Recommendations, almost all of which are also based on XML).
b) It is supported on all major platforms of the modern internet: Microsoft Windows, UNIX, Linux, MVS, VMS, OS/400, Tandem, Symbian OS and Palm OS, which cover the vast majority of devices that are of interest to this community.

### II. The language/protocol must use public-key cryptography for security.

While SKSML could be secured using one of many schemes, public-key cryptography – Digital Signatures for authentication and message integrity, and Encryption for confidentiality - is chosen for the following reasons:

a) SKSML carries a critical payload – symmetric encryption keys which will be used to protect sensitive data in the enterprise. As such, the security protecting the payload must be proven and robust. Public-key cryptography is one of the most effective means of securing an infrastructure for authentication, message integrity and confidentiality, relative to other security technologies.
b) Public-key cryptography solves all three problems – authentication, message-integrity and confidentiality – in one fell swoop, as opposed to other implementations that must choose multiple technologies to solve these three problems.
c) It is supported on all major platforms of the modern internet: Microsoft Windows, UNIX, Linux, MVS, VMS, OS/400, Tandem, Symbian OS and Palm OS, which cover the vast majority of devices that are of interest to this community.

## III. The language/protocol must NOT rely on media or transport-level security.

While the internet relies heavily on Secure Socket Layer (SSL), Transport Layer Security (TLS) and Internet Protocol Security (IPSec) for securely transporting data between two communicating applications/devices, SKSML specifically eschews transport-level security, opting for **message-level security** for the following reasons:

a) The internet consists of billions of devices and continues to grow, unabated, in types and numbers of devices. It is impossible to predict all manners and routes in which these devices may communicate based on today's implementations – they may be point-to-point between trusted entities today, but could potentially travel through many untrusted or less secure points in the future. If the message itself is "armored", then it may traverse freely and securely in any environment on the internet, including completely unsecured networks.
b) SSL, TLS and IPSec do not work in the "store-and-forward" mode of communication that must traverse unknown and untrusted network hops. Message-level security allows SKSML to be used securely in synchronous and asynchronous communication modes, over any insecure network.

## IV. The language/protocol must use existing standards where possible.

While SKSML could be based on completely new standards, the complexity of the modern internet makes the use of existing standards, where possible, a sine qua non. OASIS, the W3C, the IETF and ISO have established many standards which have established themselves through rigorous standards process and proven themselves through widespread use. Hyper Text Transfer Protocol (HTTP), eXtensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Services Security (WSS), XML Signature, XML Encryption, X.509, etc. are just some of the international standards that can, and must, be leveraged.