**Four types of web APIs**

APIs are broadly accepted and used in web applications. There are four

principal types of API commonly used in web-based applications: public,

partner, private and composite. In this context, the API "type" indicates

the intended scope of use.

**Public APIs.** A public API is open and available for use by any outside

developer or business. An enterprise that cultivates a business strategy

that involves sharing its applications and data with other businesses will

develop and offer a public API.

Public APIs typically involve moderate authentication and authorization.

An enterprise also may seek to monetize the API by imposing a per-call

cost to utilize the public API.

**Partner APIs.** A partner API, only available to specifically selected and

authorized outside developers or API consumers, is a means to

facilitate business-to-business activities. For example, if a business

wants to selectively share its customer data with outside CRM firms, a

partner API can connect the internal customer data system with those

external parties -- no other API use is permitted.

Partners have clear rights and licenses to access such APIs. For this

reason, partner APIs generally incorporate stronger authentication,

authorization and security mechanisms. Enterprises also typically do not

monetize such APIs directly; partners are paid for their services rather

than through API use.

Internal APIs. An internal (or private) API is intended only for use within

the enterprise to connect systems and data within the business. For

example, an internal API may connect an organization's payroll and HR

systems.

**Internal APIs** traditionally present weak security and authentication -- or

none -- because the APIs are intended for internal use, and such

security levels are assumed to be in place through other policies. This is

changing, however, as greater threat awareness and regulatory

compliance demands increasingly influence an organization's API

strategy.

**Composite APIs.** Composite APIs generally combine two or more APIs

to craft a sequence of related or interdependent operations. Composite

APIs can be beneficial to address complex or tightly-related API

behaviors, and can sometimes improve speed and performance over

individual APIs.