.net Magazine

Fall / Winter 2001
Vol. 1, No. 1

Get Ready for HailStorm

Microsoft's HailStorm initiative stands squarely in the middle of the Web Services movement, forming the building blocks for future Web-enabled applications.

by Jon Rauschenberger

Web Services are about to change the way you think about the Internet. Until now, most developers have viewed the Web as a mechanism to deliver an application's interface or user experience. Web Services are poised to change the way you architect and build all applications fundamentally—regardless of how they're deployed—by transforming the Web from a presentation delivery mechanism to a programming tool.

Resources

- Prototype applications on the HailStorm platform

Microsoft has announced several initiatives that will place it squarely in the forefront of the Web Services movement. Earlier this year, it unveiled a major initiative, code-named HailStorm, to build and deploy a series of Web Services that will serve as the building blocks for a new generation of Web-enabled applications. Few details of what HailStorm will ultimately look like are available, but Microsoft has released enough information to provide a good idea of what these services will do and how they might be used.

HailStorm will consist of 14 services initially (see Table 1). Users won't interact with these services directly, but will use Web sites, applications, and devices that have been built using HailStorm services (see Figure 1). Developers will build applications using the HailStorm services much as they would use SQL Server's data access services or Exchange Server's e-mail capabilities.

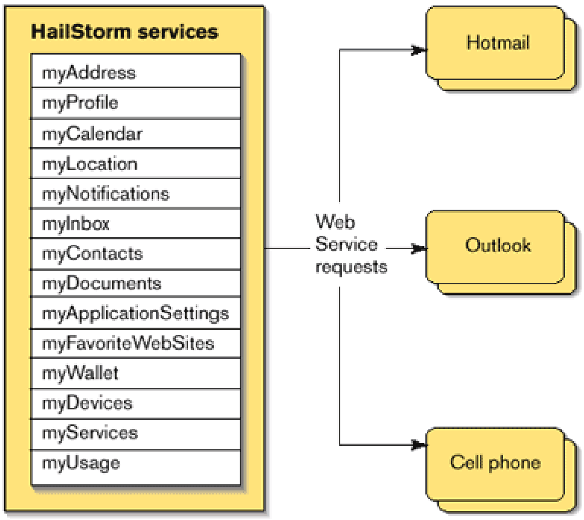

Figure 1. Program the Web
HailStorm services are accessible from virtually any language, platform, and device. This accessibility allows developers to integrate the services into a variety of applications.

Look at the myContacts service to understand HailStorm better. This service will provide address-book functionality that allows users to enter and retrieve contact information for their friends, family, or work associates. Web sites or applications access myContacts on the user's behalf using a secure Web Service interface. The contact information resides on the HailStorm servers, but the user owns the data and controls which Web sites or applications can access it.

Microsoft will probably incorporate myContacts into Hotmail. If so, users would update and retrieve addresses using the Hotmail Web site as they do today, but Hotmail would store the address information in myContacts using a Web Service exposed by myContacts.

Moving a user's address book from the Hotmail servers to HailStorm servers doesn't provide a lot of benefit for users. In fact, users might not even know when the change is made; Hotmail will continue to work as it does today. The real value comes when Microsoft updates other sites and applications to support HailStorm. A future version of Office might be HailStorm-enabled, allowing users to access the same addresses from both Hotmail and Outlook—or any Office application.

If other companies add support for the myContacts service, things could get even more interesting. If your cell phone supports the service, you could sync your address book directly to the phone without connecting it to a PC. Because the HailStorm services are built on top of Simple Object Access Protocol (SOAP), any device that supports SOAP can call them directly. Allowing users to access a single address book from their Web e-mail, corporate e-mail, and cell phone is only one of the possibilities HailStorm enables. The question isn't whether these things are going to happen; it's when they will happen and who will build the services. For its part, Microsoft plans to continue adding services to the HailStorm platform. In addition, other companies can develop their own HailStorm-compatible Web Services by following the specification Microsoft plans to release.

Microsoft Won't Own Your Data

HailStorm won't be a product you purchase and install in your data center. It will be a service running on servers owned and operated by Microsoft. Developers will code to the Web Service interfaces Microsoft defines to implement the services in their applications. These interfaces will consist of XML messages sent over a variety of protocols—most commonly HTTP. This approach raises a host of issues.

The first issue is whether Microsoft will own the data in your address book and any other information you choose to store in the HailStorm services. The reality is that Microsoft won't own the data; it will merely store the data. Users will have complete control over their data. They can choose to grant other people, organizations, or companies access to as much or as little of their data as they're comfortable with. The details haven't come out yet, but the intent is clear—Microsoft plans to treat users' data as private information and to give users control over who can access it.

Storing user information in central servers is a key aspect of HailStorm. The address-book example doesn't work unless the user's information is stored in a single location accessible from all the sites, applications, and devices the user wants to access it from.

Another issue is the level of service Microsoft will deliver with HailStorm. Any application that uses the HailStorm services will depend on the services being available. If the services go down, all applications that depend on them also go down. Microsoft is addressing this issue by building partitioning and fault tolerance throughout the HailStorm platform. It's also building a monitoring and alerting mechanism throughout the system. The company's goal is to build a platform that can deliver a level of availability, performance, and scalability that meets the demands of virtually any application. Regardless of the commitments Microsoft makes, companies need to evaluate the HailStorm platform to determine whether they're comfortable building applications that depend on it.

One challenge with using the Web as an application development platform is the difficulty identifying and authenticating users. The Web is largely an anonymous platform. Web sites usually don't know who their visitors are. Even if they can identify a user, it's difficult to authenticate that user's identity. As a result, any site that needs to identify users is forced to build its own user-authentication mechanism. Users are then required to maintain separate accounts on each site and to be authenticated by each site they visit.

The HailStorm services depend on users' identities flowing with them as they visit HailStorm-enabled sites or launch HailStorm-enabled applications. Microsoft's answer to this problem is Passport. Passport is a user authentication Web Service that serves as a key building block of the HailStorm services. Each Passport user is assigned a Passport Unique Identifier (PUID). The PUID is used throughout the HailStorm services to identify and access a user's data. The Passport service handles both identifying and authenticating users.

Microsoft created Passport originally to provide users with a single account they can use to access all of Microsoft's Web sites. Microsoft later extended it to allow any Web site to use Passport as its authentication mechanism. With HailStorm, Microsoft is extending Passport to provide authentication services for any HailStorm-enabled application.

Passport is available today. More than 150 million active Passport accounts generate a billion-plus authentication requests per month. By building HailStorm on top of Passport, Microsoft will tap into that user base. Every Passport user will have immediate access to the initial HailStorm services.

Company Expects Profits

Microsoft is attempting to define a new model for generating revenue on the Web with HailStorm. HailStorm will be a fee-based service that consumers pay to use. The pricing details haven't been released, but the plan is for Microsoft to bill customers directly for the services they elect to use. Microsoft also will charge fees for companies that implement the services in their applications, sites, and devices, but these fees are supposed to cover only the support costs Microsoft incurs with each organization.

Microsoft plans to operate HailStorm as a separate business unit. This means this business unit needs to generate at least enough revenue from the services to cover its operating costs. This approach is different from many of Microsoft's past online ventures, which were designed to build brand awareness, not to generate a profit.

Many users have come to expect services such as HailStorm to be free, so Microsoft needs to deliver a compelling set of services to convince users to pay for using them. Microsoft also needs to generate broad third-party adoption of the services. If users can access the HailStorm services from a variety of sites, applications, and devices, they will be more likely to pay for those services.

The Business Case

Passport to a Better Bottom Line

To see how future Web Services will impact your bottom line, take a look at Microsoft Passport, the basis for user identification in future HailStorm services...

Click here for more

A subset of the HailStorm services probably will be available at no fee. However, Microsoft will push to convince users to upgrade to the fee-based services. A service such as myNotifications might act as the "killer service" to convince many users to upgrade. You might see a site such as Hotmail continuing to offer free e-mail services using the myContacts and myInbox services, but if users want access to richer functionality using the myNotifications and myCalendar services, they will need to pay for those services.

Tech Tip

Use Value Types for Performance

In .NET, everything is an object and derives from a common base class. That's great for flexibility, giving you advantages such as runtime type checking and type conversion...

Click here for more

Several major companies have announced plans already to support HailStorm. eBay will use the myNotifications service to alert customers to the status of their bids and auctions. American Express will use myWallet and myNotifications to simplify online purchasing and alert customers of possible fraudulent activities. eBay, American Express, and several other companies have built prototype applications already on the HailStorm platform (see Resources).

Third-party developers have several compelling reasons to add Passport/HailStorm support to their products. Passport is the most widely used authentication service on the Web today. Supporting Passport authentication makes it easier for users to tell you who they are, and it removes one of the most common barriers to entry or purchase—the need to set up a new account.

The HailStorm services allow developers to build functionality into their systems that's not possible today. Initially, making applications HailStorm-enabled allows developers to differentiate their products from competitors. Over time, users will probably grow to expect the kind of functionality

HailStorm enables, and developers will have to find new and creative ways to leverage the services to serve their customers better.

It's clear that the Web as we know it is poised to change dramatically. Companies are looking for ways to use the Web as more than a content-delivery mechanism. They're looking for opportunities to help run their business more efficiently and to enable new business opportunities. Web Services will provide the foundation necessary to facilitate this change.

With HailStorm, Microsoft has taken the lead in building broad-reach Web Services. However, being the first to enter this market doesn't guarantee success. Microsoft faces a number of challenges, both technical and nontechnical. However, it's uniquely positioned to succeed in this area. The combination of the existing Passport user base, the Microsoft Network (MSN) collection of Web sites, and Microsoft's desktop platforms and applications gives the company access to all the pieces necessary to build a compelling offering. If that offering can deliver on the promise of Web Services and address security and privacy issues, HailStorm might change the way you build your Web-enabled applications.

About the Author

Jon Rauschenberger is a partner and director of technology at Clarity Consulting Inc., a Chicago-based information technology consulting firm and Microsoft Gold Certified Partner. In addition to architecting and building scalable Web-based solutions, Jon speaks at conferences such as Microsoft Tech•Ed, VBITS, COMDEX, and DevDays. He is also the MSDN regional director for Chicago. Reach Jon at jrausch@claritycon.com.

Service	Description
myAddress	Electronic and geographical address for an identity
myProfile	Name, nickname, special dates, picture
myContacts	Electronic relationships/address book
myLocation	Electronic and geographical location and rendezvous
myNotifications	Notification subscription, management, and routing
myInbox	Inbox items such as e-mail and voice mail, including existing mail systems
myCalendar	Time and task management
myDocuments	Raw document storage
myApplicationSettings	Application settings
myFavoriteWebSites	Favorite URLs and other Web identifiers
myWallet	Receipts, payment instruments, coupons, and other transaction records
myDevices	Device settings, capabilities
myServices	Services provided for an identity
myUsage	Usage report for preceding services

Table 1 **HailStorm Services** HailStorm, consisting of 14 services initially, will enable a new breed of applications that give users access to their data from any site, application, or device. Users will use Web sites, applications, and devices built using HailStorm services, but they won't interact with these services directly.