

# Beyond IP

## Taking Access Control Beyond Its Boundaries with XML

WHITE PAPER



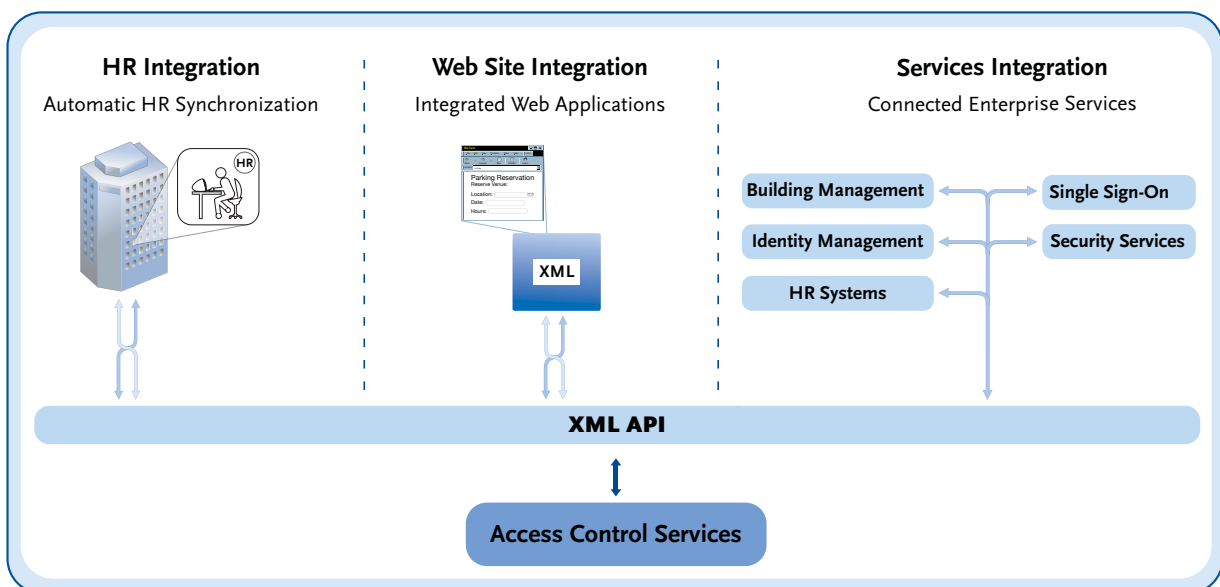
## Introduction

The days of security systems operating in isolation from other IT systems are over.

Customers are increasingly demanding that security system components interoperate with each other, with their IT systems, and with a host of other systems ranging from HR to building automation. They are not content to use outmoded, patchwork technologies to accomplish this, rather, they are insisting on compliance with industry standards, open interfaces, simple tools, and network-friendly, IP-based solutions that leverage existing resources to their best advantage.

That's why all Brivo Systems' products support an XML-based Application Programming Interface (API), a tool that allows them to interoperate easily with other IT and security systems.

Using the XML markup language for data exchange, integrators can connect access control systems to virtually any other information system that supports the HTTP protocol, regardless of underlying hardware, operating system, or programming languages.



**Figure 1: Access Control Services Integration**

## What does the API do?

An API is a set of functions that one computer program makes available to other programs so they can talk to each other directly. Put another way, it is a set of tools for programmers that allows one computer system to use the services offered by another.

A contemporary example would be online services such as Google Maps™ or Mapquest®: both provide APIs that work behind the scenes to allow your company's website to invoke dynamic mapping capabilities from either of these online services. Your company's website is using services provided by a mapping company's Web server through an API.

The types of services provided by an API depend on the type of business that designs the API. For example, a consumer-oriented business like Amazon provides a "retail API" that allows third parties to set up their own storefronts, with Amazon processing transactions in the background. Likewise, an online access control

company like Brivo offers an API that provides a range of services for managing physical security. Both examples use the same underlying technologies, but for different business purposes.

What are the services that Brivo provides through its APIs? In broad terms, they can be described as:

- Synchronizing user identity data with other applications and databases
- Controlling behavior of access control equipment and connected physical devices
- Receiving real-world event notifications detected through Brivo-connected sensors
- Forwarding of alarms, security violations, and business rule triggers

We call this collection of services our Service Oriented Security Infrastructure™ (SOSI).

## Service Oriented Security Infrastructure™ (SOSI)

Our SOSI framework is based on Service Oriented Architectures (SOAs). The concept of SOAs, while fairly simple, has been exerting a tremendous transformative force on the IT world in the years since widespread, IP-based connectivity has become the norm. Within the security industry, Brivo has been at the forefront of this trend, providing access control and related security functions as a service available over the Internet for more than six years.

### Service Oriented

What does it mean to have a service-oriented architecture or API?

At its core, the SOA viewpoint on system design says that collaborative software should present an à la carte menu of simple operations—services—that other systems can use individually or in combination to accomplish a specific task. In this way, an integration project doesn't have to bite off a whole chunk of data modeling or programmatic interfacing to perform a simple task; it can select only the pieces it needs.

Just want to add a user? No problem. Just learn one XML format and submit it through a secure HTTPS transaction. You're done.

Delete that user? A few more lines of XML, the same secure HTTPS transaction, and that's it.

The same is true for scores of other transactions that you might want to conduct with an access control system. Each is presented as a discrete service that will perform a specific task for another IT system. What could be easier?

That's service-oriented.

### Security Infrastructure

What does it mean to be part of the infrastructure?

The "security infrastructure" aspect of our XML offering reflects the growing trend toward designing security products not as standalone systems, but rather as individual parts that can be combined to form an overall corporate infrastructure that includes many components from many sources.

That's because in today's complex integrated architectures, at any time a particular subsystem might need to play a leading role, or simply become part of the infrastructure.

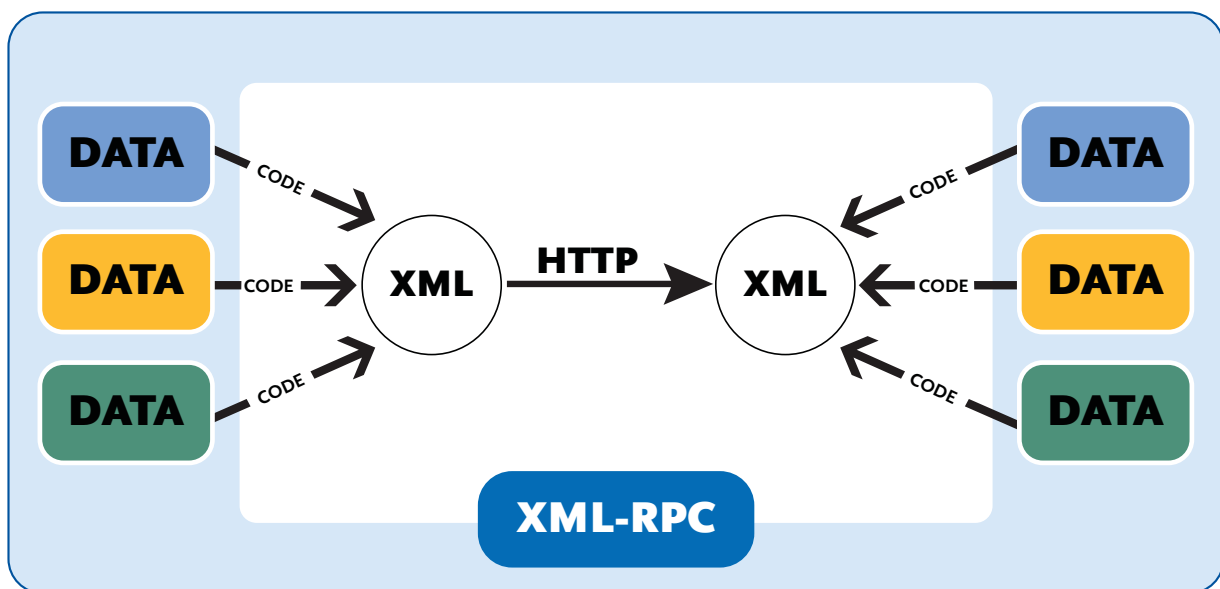
In a leading role, a subsystem might be the primary point of interacting with end users, and might then be responsible for command and control of other subsystems. Many customers use Brivo this way when they tap into our video integration services. These customers interact primarily with our own browser-based GUI, which in turn commands a DVR to display the footage they want to see associated with an event. All this is accomplished without ever leaving the primary Web-based Brivo application.

On the other hand, a subsystem in an infrastructure role would not be the primary point of interaction with end users. Instead, it would receive commands from some other system, and simply execute those commands or supply requested data. In this case, customers who have integrated an HR front-end with their access control system, for example, are using the access control system in an infrastructure role. The system is there in the background performing a valuable service for the rest of IT architecture through open, standards-based APIs.

That's security infrastructure.

## What Is XML-RPC?

XML-RPC is an example of a Web service protocol. It uses XML for data formatting and the HTTP protocol for data transport. We've used these open standards to implement the services offered through our published APIs.



**Figure 2: XML-RPC Data Exchange**

## XML

XML is by now a familiar term to most people in technology-related industries. That said, it carries different connotations for different readers, so it's probably worth a quick review, particularly in terms of its implications for integrators or developers who would like to work with Brivo's XML-RPC API.

Like the more familiar HTML used to format data for presentation on Web pages, XML is a "markup language" that uses "tags" to format, or mark up, data. (Tags are the words enclosed in the "<" and ">" characters, as shown in figures 3, 4, and 5.) In the case of XML, however, the formatting is intended for other computers and software programs to read—not for humans to read from a Web page. The process of using XML tags to mark

up data is intended to let another computer system understand the meaning of the data being presented and take appropriate action as called for by the particular application of the technology.

One of the primary virtues of XML is that it is text-based, and, as such, is independent of programming languages, operating systems, development environments, and underlying computer or processor design. That's what makes it such a great choice for integrating otherwise disparate systems, and why it has enjoyed such popularity on the Internet, where systems of all types must be able to communicate freely.

Bringing XML into the security industry through our API is how we support broad interconnection between many types of applications, hardware platforms, operating systems, and device types.

## XML-RPC

As defined at [www.xmlrpc.com](http://www.xmlrpc.com), XML-RPC is a specification that allows software to run on disparate operating systems, as well as in different environments, to exchange structure data over IP networks. Thus, XML-RPC is nothing more than a remote procedure call using HTTP as the transport and XML as the encoding. XML-RPC is designed to be as simple as possible while allowing complex data structures to be transmitted, processed, and returned.

We have selected this mechanism due to its simplicity, its openly available implementation on multiple operating system platforms, and its compatibility with the information security requirements and environments commonly found in our end user premises. XML-RPC is very IT-friendly because its networking requirements are about the same as those of a browser running on a desktop PC. Because most corporate networks are configured similarly in the first place, the use of XML-RPC in one of your applications will probably not impose any new information security requirements or procedures.

What's also IT-friendly is that the same API is available on all of Brivo's offerings so that developers can use a single API regardless of the target Brivo product. No rewriting of code required.

## Applying XML-RPC to Real-world Systems

How does XML-RPC apply to real systems as part of a security infrastructure?

The reference model figure 3 on the following page shows the major components that we will be focusing on to answer this question. Starting at the left of the figure, there is an external system that wants to use the services offered by a core software system of some sort. It does so by exchanging XML-formatted messages over a TCP/IP network, specifically using the HTTP protocol. This exchange is possible because the core system has a published Web services API that uses XML-RPC.

### Example: Adding a User

In the context of security infrastructure, the reference model can be used to illustrate an exchange that populates the access control system with a user profile from some external IT system, such as the Human Resources (HR) system.

Figure 3 provides a simplified example of an XML-formatted message being transmitted via the XML-RPC mechanism. The message indicates the name of the user, a card number, group membership to establish permissions, and a schedule to constrain the time period for those permissions.

At the successful conclusion of this exchange, the user has been added to the access control system and enjoys all the access privileges of the “Staff” group. Through the use of XML, we have reduced two manual data entry tasks down to one automatic transaction.

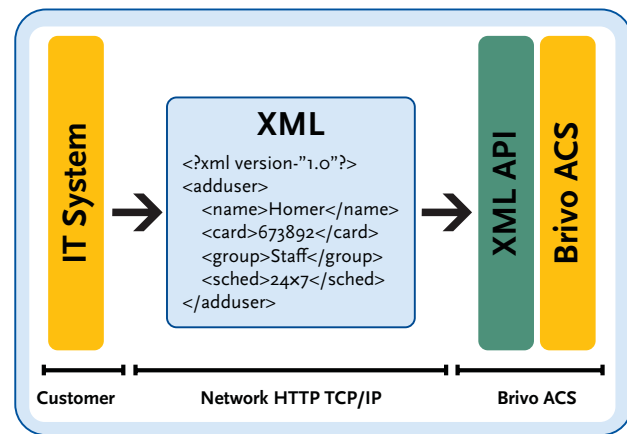


Figure 3: Add User via XML-RPC

### Example: Deleting a User

Upon termination of employment, this user will need to be removed from the access control system. This can be accomplished through a similar XML exchange between the external IT system and the access control system.

In figure 4, the user is being deleted with a simplified version of the command that uses the name or other unique identifier that represents the individual to be deleted.

At the successful conclusion of this exchange, the user has been removed from the access control system and no longer has any privileges.

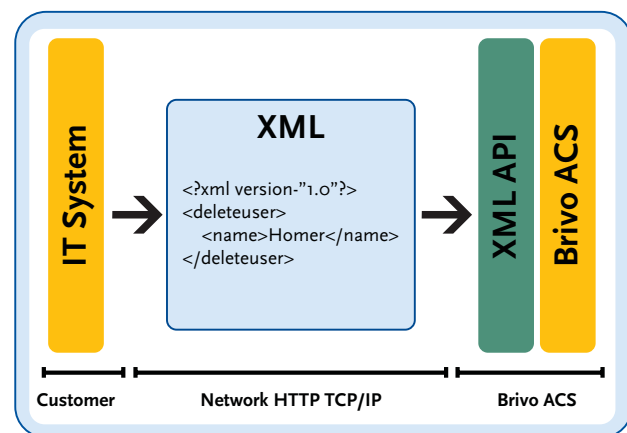


Figure 4: Delete User via XML-RPC

### Example: Subscription Services (Notifications)

So far, both real-world examples have shown transactions initiated by an external IT system. What if, alternatively, a system needs to be notified when an event or alarm of a particular type occurs, rather than initiating the transaction itself? Is that possible with XML-RPC?

Fortunately, XML-RPC can be used in both directions in Brivo's SOSI architecture by interchanging roles with the external IT system.

Ordinarily, a software system with an XML-RPC API runs a Web server that accepts XML documents (commands, queries) over HTTP and then processes them accordingly. By running its own Web server, any external system can receive XML messages in exactly the same way, using all the same software tools.

Figure 5 depicts a simplified “door ajar” alarm message transmitted from an XML-RPC to the IT system, a system equipped with its own Web server capable of receiving and processing XML messages.

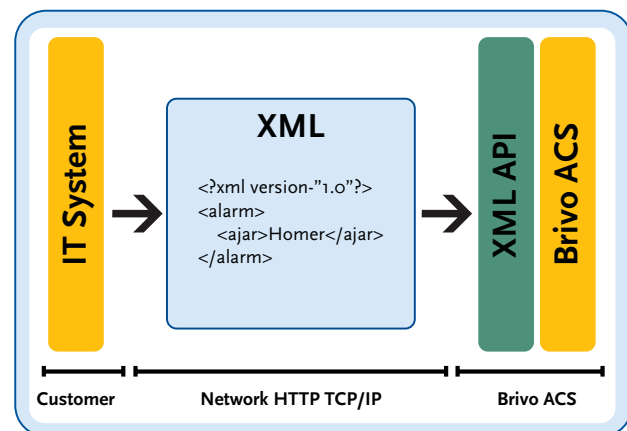


Figure 5: Alarm Message via XML-RPC

## Case Studies

Now that we've seen what some individual transactions look like, we can broaden our view to look at how these pieces can be put together to create new business value beyond the boundaries of traditional access control.

### Hospitality

The hospitality industry must deliver dependable, quality services to guests, and information systems are an essential part of delivering those services reliably. But what happens when that's not possible? The company will see its costs increase or more likely incurs a loss of revenue as guests become dissatisfied, resulting in headaches for everyone.

In this example, a resort management company with hundreds of rooms spanning multiple properties has a problem delivering parking reservations to guests through its network of unaffiliated rental agents. With no way for rental agents to guarantee them a space in advance, guests cannot find parking for their cars and they become frustrated. And with no infrastructure to manage parking assets, the property owner is missing a major revenue stream.

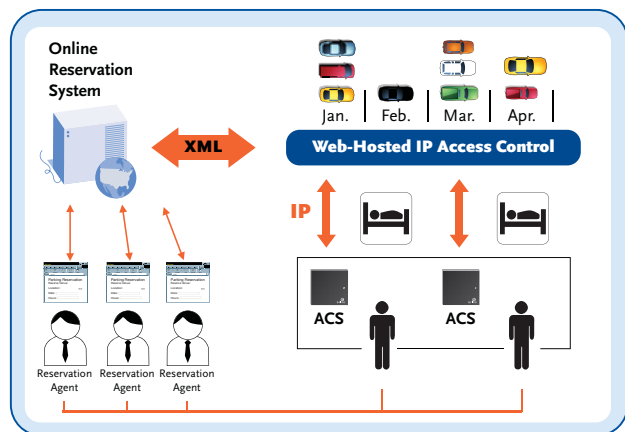
In a nutshell, their problems include:

- Not enough parking spaces to meet guest demand
- Unaffiliated nationwide rental agents with no shared information system
- No way to forecast demand or reserve parking
- Insufficient data to invoice for parking
- Unhappy guests who cannot count on parking availability

A creative XML solution put together by our Custom Programming group solves all of these problems with a new website for online reservations. Built as a free-standing application outside of Brivo's core ACS WebService system, the new website presents a booking front-end to the application that will allow rental agents to guarantee parking for their guests. The back-end of the application uses our XML interface to translate reservations into access privileges for specific garages on specific dates and times using traditional proximity card technology.

Rental agents, regardless of their location, can now book parking spots along with accommodations via the Internet. Because the reservation system manages available spots for up to a year in advance, parking is no longer overbooked and guests always get what they requested in the reservation process. Normal access control schedules on guests' parking passes expire when their reservation expires. Once a guest checks out, even if they keep their card, it will no longer permit entry to the facility's garage.

In this way, parking is controlled not only to provide convenience to current guests, but also to open up an additional revenue stream for the property management company. By being able to guarantee parking place for the guest's entire stay, the resort can now charge for the spot, with the online system providing the detailed records needed to drive the invoicing process.

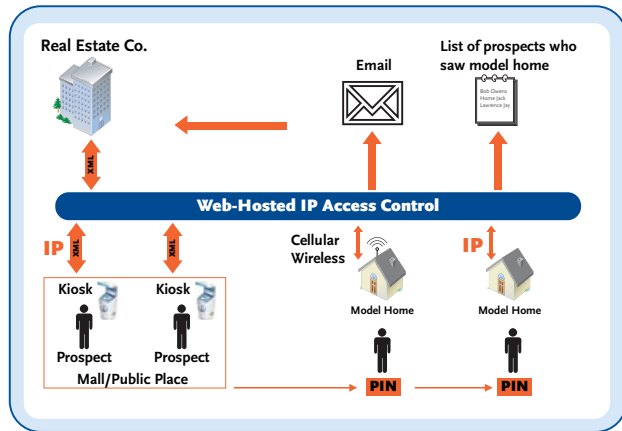


**Figure 6: On-line Reservation via XML-RPC**

## Real Estate

Successfully reaching interested buyers is essential for any business. Wasting time with the wrong prospects is expensive and non-productive.

Here again we see the geographically distributed point-of-service configuration that Web-hosted access control serves so well. This scenario also details the problem encountered by kiosks located in shopping malls that allow prospective real estate buyers to register and then visit properties offered by the real estate company. Without being linked to a central system, each kiosk's information must be manually retrieved and there is no tracking of actual visits, no way to know who is visiting, and possible sales leads cannot be tracked. There is also a need to obtain valid credentials for visitors so that they may gain access to a property—if an agent is not present to meet the visitor, or the property would have to be left open in order to allow the visitor access.



**Figure 7: Real Estate Application via XML-RPC**

The XML interface links the kiosks to the access control system and, in turn, to the individual properties. Wireless control panels at each property avoid the need for special wiring. The visitors are able to register at the mall kiosks and immediately receive a Personal Identification Number (PIN) that will grant them access to the desired property. The PINs can be enabled to be valid only during certain times, configured to avoid multiple visits, and prevent sharing of credentials with non-registered visitors.

The added value for the real estate company is having a detailed record of who actually visited the property. Follow-ups are now highly efficient and targeted, leading to better closure rates and more productive agents.

## Getting Started

Getting started with XML-RPC is easy because it's an open standard. You don't have to buy anything from anyone to start using it. However, you will need some tools to help formulate and interpret XML messages, and to send them using XML-RPC.

### Computer system requirements

Of course, you'll need a computer system capable of supporting XML Web programming and Internet access. These days, just about every computer has these features, so you can probably get started with equipment that you already own. Almost any operating system will do—Windows, Linux, Unix—and the links to the requisite XML-RPC libraries are available online for all of these operating systems—for free—as described below.

### Downloading an XML-RPC Implementation

If you want to start working with this technology, you'll want to go to visit [www.xmlrpc.com](http://www.xmlrpc.com). This official site of the XML-RPC protocol provides numerous resources, including a set of links that will take you to sites that offer downloads of various XML-RPC implementations.

What's an implementation? An implementation is an SDK (Software Developers Kit) that provides the tools to build XML-RPC interfaces in a particular operating system and language. The SDKs you'll find at these sites

are completely generic and have nothing to do with the security applications discussed in this paper. They're simply tools that you can use for any XML-RPC project.

These are available for .NET developers as well as many Linux/Unix implementations.

As for languages, implementations can be found for C/C++, Java, PHP, Perl, and many others.

This all adds up to a tremendous level of flexibility with regard to the programming options of how external systems will interact with open systems like Brivo's.

## Brivo Developer Program

Brivo makes its entire Software Development Kit (SDK) available with a simple Terms of Use agreement. Write to [api@brivo.com](mailto:api@brivo.com) and ask for a copy.

Our specification (SDK) lists all the XML tags we use, what data you can exchange with us, code samples, and how to authenticate your application (log in) to a Brivo system.

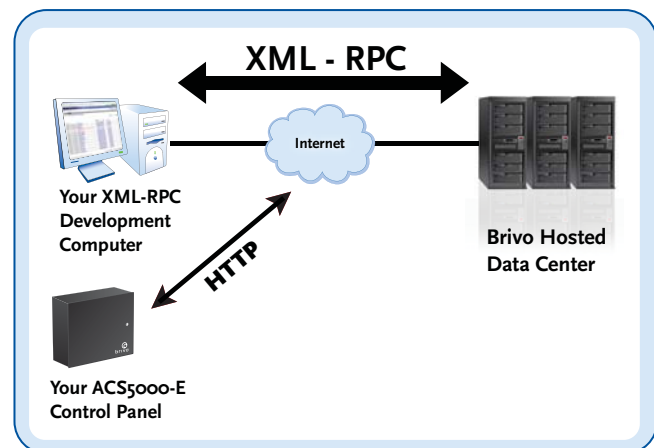
The only requirement of the developer program is that you agree to a brief "Terms of Use" agreement. It's not a license or a non-disclosure agreement—nothing quite that legal. It's simply a means for us to protect all of our customers from potential API abuses by anyone else using the system.

Even if you are only "kicking the tires" or trying to evaluate whether the API is right for you, we invite you to join the developer program. It's a way for us to keep in touch with you so that you'll always know about updates and improvements to specifications and capabilities.

## Accounts and Control Panels

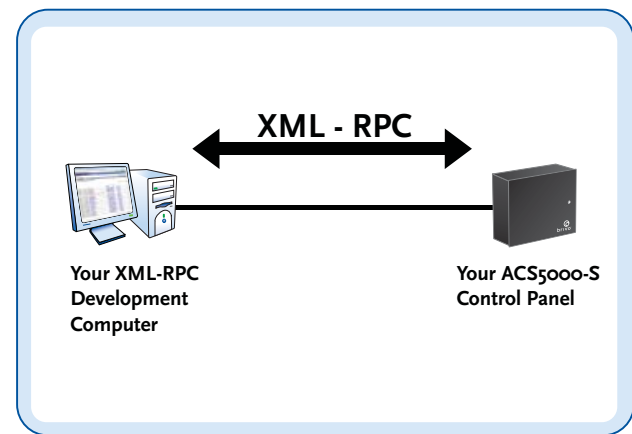
To get an actual project started, you will need access to a real control panel in order to fully test your integration project. There are several ways to do this.

If you are already a Brivo customer, you can contact our Technical Support department and ask them to activate an "XML Logon" for your existing account. This will let you access your current control panel infrastructure through our ACS WebService product, available via the Internet. However, we don't recommend that you develop software against a control panel that's providing critical security services for your building. This type of development would best be kept to a control panel that's on the bench or at least not critical to your outside security.



**Figure 8: XML-RPC via ACS WebService**

You can also interface directly with a Brivo ACS OnSite control panel right on the bench. This is possible because our API is the same whether you're connecting to an ACS OnSite panel over your own LAN or connecting to an ACS WebService panel over the Internet. Using an ACS OnSite panel for development is a great option for maintaining control and not disrupting operations while developing software. ACS OnSite products are available for purchase through our dealer network.



**Figure 9: XML-RPC via Local LAN**

## Contact Information

Brivo recognizes that many customers prefer to focus on their own core business competencies, and leave the computer programming to others.

To meet that need, Brivo now offers custom programming services based on our own XML API. If you would like to explore a special requirement or to explore these topics further, please visit [www.brivo.com/dealers/professional-services](http://www.brivo.com/dealers/professional-services) to submit a request detailing your project.

We look forward to working with you.

© 2007 Brivo Systems LLC. All brand and product names are trademarks of their respective owners.