



Mobile Enterprise Solutions

W H I T E P A P E R

.NET Technology & Service-Oriented Architecture:

*What they are, how they work, and
how they deliver business value*

RF-SMART® is a division of ICS, Inc.
1650 Prudential Drive, Suite 300
Jacksonville, FL 32207 USA
+1 (904) 399 8500 www.rfsmart.com

About .NET Technology

The .NET framework is a powerful, globally-accepted development platform and Web services communication strategy. It enables businesses to integrate systems more rapidly and in a more agile manner to help them realize the promise of information anytime, anywhere, on any device — regardless of the underlying platform.

As a framework for building, deploying, managing and using service-based applications, .NET is remarkably scalable and provides a significant amount of “behind the scenes” code. This enables measurable productivity advantages in solution development and maintenance...two important factors why .NET has perhaps the widest level of use.

UNIVERSAL SUPPORT, LOW TECHNOLOGY RISK

.NET is *substantially* well-supported by its developer (MS), more than 35,000 vendors around the world, and a global community of programming talent. This level of global acceptance not only validates the performance and usability of .NET, it also means that resources (coders, tools, user groups, conferences, etc.) are available anywhere in the world.

The lifespan of the base code, its degree of mainstream acceptance, and the availability of support resources are pivotal considerations for mission-critical applications. While .NET offers an unparalleled degree of confidence that the language and tools will exist longer than the software being purchased, custom applications written with code that has failed to gain mainstream acceptance dramatically increases technology risk and severely limits support/resource availability. Sooner or later—*but certainly*—the cost of running “dead” technology outweighs the application benefits and the cost of moving forward. The technology risk with .NET and RF-SMART is virtually zero.

.NET PERFORMANCE

Server applications have never been faster—or more scalable—than with the .NET framework and its ASP.NET technology. .NET’s native XML approach to data access frees up database connections and results in significantly greater scalability. It is not uncommon for companies to experience improvements in application speed on the order of 200% to 400%...using the same hardware. And, one installation of RF-SMART on a properly-equipped server (PC) can support up to 1,000 RF-SMART users.

EASE & COST OF OWNERSHIP

.NET is a very intuitive, low-overhead programming model. With the .NET Framework, an application can be transformed into a Web service with just a few simple lines of code. Why?

The amount of code provided in .NET class libraries and the amount of work that the .NET framework handles behind the scenes allows developers and system administrators to reap huge productivity gains in building and supporting applications.

BUILT-IN SECURITY ADVANTAGES

The code access security technology in the .NET Framework was designed for today's Internet environments. The .NET framework can collect evidence about the origin and author of an application then combine that evidence with administrator-set or default security policies to make fine-grained decisions about whether (or how) to interact with the called application. This is standard with RF-SMART.

RF-SMART supports Active Directory and Windows security for user validation. Our customers can manage all user names and passwords for ADC, ERP and Windows in one central location. RF-SMART security rules not only mimic and reinforce ERP rules, they offer a more granular level of security (Example: Bob can do positive but not negative inventory adjustments). And, all RF-SMART functions feature role-based, read-only security to safeguard access to ERP data.

About SOA: Service-Oriented Architecture

HOW SOA EVOLVED & WHY THAT'S IMPORTANT

Early on, programmers realized they needed a better way to reuse some of the complex code that they were writing and rewriting. The "cut and paste" approach allowed hard-coders to reuse certain subroutines and functions, but provided no way to maintain that code other than to track down every instance in which the code was used, fix the code in each instance, then re-deploy each instance *and* the whole system. It remains a maintenance, deployment and cost nightmare.

The next generation of developers moved on to *object-orientation* and *component-based software* models. These approaches enabled coders to reuse and maintain code and *functionality* (allowed them to fix the code in one place). While these approaches are vastly superior to cutting-and-pasting hard code, they fall short in addressing today's more abstract IT complexities.

Today, developers must efficiently address more abstract concepts like the Internet, distributed software environments, integration with multiple applications and OS platforms, support for a slew of communication protocols, ERP migrations, network security, support for various devices/clients, and perhaps most importantly, provide the ability to quickly adapt.

SOA provides the solution to *all of the above* by eliminating the headaches of:

- Writing, maintaining, and redeploying hard-coded applications and functionality
- Supporting multiple, contemporary, open-standard communication protocols (XML, SOAP, HTTP) with *little to no* programming
- Seamlessly and efficiently integrating with multiple applications (ERP, manifest systems), various OS platforms, and "intelligent" pieces of equipment (conveyors, carousels)

Interestingly, these are the promises of Fusion. RF-SMART is already there, today, out-of-the-box.

NON-TECHNICAL SOA DESCRIPTION

A service-oriented architecture (SOA) lets any application correctly, securely and dynamically talk to any other application. It does this by allowing specific, independent application components (“services”) to communicate with each other. This can be as simple as passing along data or it could involve two or more services coordinating some activity. It is the combination of services—internal and external to an organization—that make up a service-oriented architecture.

How specific services interact is determined by *messages*. In layman’s terms, a message is like a “personal ad:” It describes and identifies itself (security validation) and tells others what it’s looking for (specific data or process). Messages are typically conveyed using HTTP and XML over computer networks...including the Internet.

The way to fully realize the benefits of SOA is to use *Web services*. A Web service is basically an open-standards way to allow different applications (on any platform, any network) to interact in a manner similar to the inter-process communication on a single computer. It’s *that* efficient. In a nutshell, Web services are the ultimate interoperability enabler.

MORE TECHNICALLY...

SOA is an application architecture in which all functions are defined as independent, discoverable “services” with distinct, evocable interfaces (actions) which can be called in particular sequences to form business processes.

Because the calls and sequences are simple to configure, SOA enables organizations to wield a more powerful, flexible programming and application-communications model that reduces development costs, ownership costs, and implementation risk.

As a “real-time, dynamic data bridge” between different systems in different places on different networks, a service orientation hides the complexity of today's heterogeneous IT environments and enables a level of IT adaptability that has never before been possible.

From a business value perspective, SOA enables:

- A reduced reliance on expensive custom development and hard-coded interface maintenance
- Use of existing investments in technology (ERP, legacy systems, middleware, etc.)
- Better security—SOA has authentication and authorization support at every level
- Lower costs of new technology integration and maintenance
- The ability to more quickly meet (adapt to) changing customer, vendor and regulatory demands
- Business functionality to be managed closer to the business units

THE ROLE OF WEB SERVICES IN SOA

Web services—which are built on top of well-known, platform-independent protocols like HTTP, XML, UDDI, WSDL and SOAP—play a major role in fulfilling the key objectives of SOA applications:

- UDDI, WSDL and/or SOAP allow services be dynamically discoverable and evocable

- XML enables a scalable, robust, platform-independent interface contract
- HTTP fulfills the promise of interoperability

It is the combination of these protocols that make Web services so attractive and the reason SOA is the architecture of choice for developing responsive, adaptive, composite applications. Customers can even plug-in customer/partner workflows without any effort from ICS.

This makes RF-SMART a true enabler for enterprise-wide data accuracy, visibility and connectivity.

The Potent Combination of .NET & SOA

The .NET framework is a powerful service-oriented *environment* for building, deploying, managing and using service-based applications. If a technology foundation could be compared to a house, the .NET framework provides the foundation (code), the framing, plumbing, electrical, security and communications systems...all of which work together, are expandable, configurable, and are easily connected with any other house or business. Plug new things in and they work. *It's that cool.*

Service-oriented *architecture* is the best strategy yet for delivering scalable, high-performance business agility, IT flexibility, and low cost of ownership.

Not only does RF-SMART feature an SOA, it is built using tools designed for optimizing the performance of SOA.

RF-SMART Technology Summary

- Superior, globally-embraced code and tools designed for today's complex IT needs
- Substantially more prudent & advanced architecture (SOA vs. cutting-and-pasting hard code)
- Geometrically—perhaps exponentially—greater scalability
- More flexible and responsive technology environment...you're ready for anything
- Significantly lower cost of technology and solution ownership
- Low-to-no technology risk. .NET is substantially well-supported on all fronts.
- Tighter, more granular security controls
- Incredible configurability without programming
- RF-SMART serves to *unify* an organization's technology stack...not further *diversify* it

PLEASE SEE THE FOLLOWING PAGE FOR QUOTES ON SOA

What Experts Say About SOA

[Aberdeen Group. SOA In IT – Benchmark Report. January, 2006](#)

“Over the next five years, Global 2000 companies [alone] will save up to \$53 billion from their IT budgets as a result of SOA initiatives.”

[Sayed Hashimi, MEng. Independent technology consultant and research professional specializing in languages, algorithms, integration and mesh generation. 2004](#)

“Application integration is one of the major issues companies face today; SOA solves that. System availability, reliability, and scalability continue to bite companies today; SOA addresses these issues. Given today's requirements, SOA is the best scalable solution for application architecture.”

[Aberdeen Group. SOA in the Supply Chain – Benchmark Report. September, 2005](#)

“SOA represents the next big breakthrough in software design and deployment, leading to cheaper and faster integration and more flexible business processes.”

“SOA’s immediate promise is to significantly reduce the cost and complexity of application integration, which can be as high as 40% of a typical IT budget.”

[Forrester Research. SOA Will Change How IT Works. May, 2005](#)

“The flexibility provided by SOA enables the continuous optimization of business processes. But the traditional IT organization, which is oriented toward discrete business units and supported by vertically integrated applications, constrains this optimization rather than helps.”

[Oracle Corp. White Paper: Oracle Fusion Middleware 10g R2 Oracle Enterprise Messaging Service. November, 2005](#)

“Many information technology departments [have] deployed enterprise applications using a fragmented, piecemeal middleware infrastructure. The resulting complexity represents nearly 50% of the information technology costs and 60% of organizations consider their infrastructure an impediment to their ability to meet business requirements. Enterprises are evolving their applications from being monolithic, closed systems to being modular, open systems with well-defined interfaces. This is service-oriented architecture (SOA).”

[Davis Media Custom Publishing. Strategies for SOA Success—Sponsored by Oracle. December 2005](#)

“Embracing a service-oriented architecture is crucial to any company hoping to build a competitive edge....That’s because SOAs directly address the many hurdles that IT departments must overcome in order to deliver on the dynamic and rapidly evolving requirements of the business.”

[Microsoft. White Paper: Service Orientation and Its Role in Your Connected Systems Strategy. 2004](#)

“To the chief information officer, service orientation is a means for protecting existing IT investments without inhibiting the deployment of new capabilities. By encapsulating a business application behind capability-based interfaces, the service model allows controlled access to mission-critical applications, and creates the opportunity for continuous improvement of the implementation behind that interface. Service orientation protects investments from the swirl of change.”