



## Bridging the SOA divide for deployed assets

*How the Feds must rethink software design principles to  
better serve users abroad*

By [Cameron Matthews](#), CTO  
Sentek Consulting

Service-Oriented Architecture (SOA), the definition of grouping functionality around business processes and packaging them as interoperable services, is ideal for most government and military technology environments. This practice allows different applications to exchange data with one another by loosely connecting services with operating systems, programming languages and other technologies that underlie applications. These services communicate with each other by passing data or coordinating an activity with other systems. SOA affords agencies the ability to take advantage of new technologies and respond to end-user demands more quickly and cost-effectively.

However, establishing and maintaining connectivity between applications and services can be significantly difficult in forward-deployed operations. Military commanders in the Northern Mountains of Afghanistan may find satellite coverage spotty or conditions unsafe to stop, upload and share information. State Department staff and U.S. Agency for International Development members may not have access to

dependable power or telecommunications in areas like West Africa as they seek to learn the latest whereabouts of warring tribes seeking to steal their humanitarian goods and services. Justice, Homeland Security and other federal intelligence agencies may also find themselves seeking information about critical leads and tips in the jungles of Southeast Asia, but lack stable Internet connectivity to communicate with the outside world.

### Infrastructure issues abound

These scenarios are becoming more commonplace as the Global War on Terror requires more U.S. government and military personnel to operate in remote areas for longer periods of time with little - if any - reliable infrastructure to support operations. Forward-deployed assets will undoubtedly deal with regular power outages,

sporadic Internet connectivity and high-packet latency.

Furthermore, initiatives such as the U.S. Navy's Consolidated Afloat Networks and Enterprise Services, or CANES, will not only bring about faster implementation of new technology aboard ships, aircraft and ground units at a reduced cost, but will also bring to the forefront the need to service the small, but active number of forward-deployed military personnel who are dealing with significant geographic and communication limitations that make them ill-suited to share and transfer data in large amounts on a continual basis.

### Rethinking data exchange strategies

This doesn't mean that all government SOA strategies should stop. Far from it; its benefits are too good to not implement. However, there will need to be some rethinking by program managers and engineers about how data exchange can occur with units and individuals operating in remote areas. Here are just a few examples:

## Resolving the “Small Pipe Syndrome”

For issues where remote operations are preventing forward deployed assets from downloading large amounts of data before timing out, administrators should consider implementing policies and practices to move data along, such as:

- Using compression software on SOA Web service call payloads or transport-level compression (e.g. GZIP over HTTP)
- Reducing the amount of Web service calls and messages that are sent.
- Batching information together where possible to reduce call overhead.
- Using longitudinal communication encoding (never sending the same full message twice).
- Designing the system to not use polling to detect updates if they occur elsewhere or on the server in order to avoid clogging the network.

- Designing the system to be event-based with server push to reduce the update detection traffic to a minimum.

## Overcoming intermittent connectivity

To better serve end users operating in locations with unstable or unreliable power and Internet access, agencies may want to employ these tactics:

- Using a GUI client package with Web service access to develop the SOA backend.
- Designing both client and server to be event-oriented, and evaluate an overall Event-Driven Architecture for its proper fit.
- Designing both the client and server to be asynchronous message-

based rather than using synchronous Web service calls, thus preventing either side from hanging up when a connection isn't possible.

- Pushing as much intelligence as possible onto the smart client, since it is closer to the user and will retain functionality even if the connection drops.
- Queueing up all messages to be delivered when the connectivity returns.
- Understanding the issues with cache coherency and reconciling differences once the connection is regained.

The examples illustrate how traditional SOA engineering assumes the basic infrastructure requirements are available and accessible. Many federal agencies will tell you that such is not the case for some of their units operating at the tip of the spear. Ironically, these individuals have a great need for real-time information to conduct their mission. SOA-related initiatives need to respond accordingly.

*About the Author: Cameron Matthews is the CTO for [Sentek Consulting](http://Sentek Consulting), a rapidly-growing provider of government and commercial IT security and C2 programs, including security, program management, strategic consulting, engineering, software development and acquisition support. He can be reached at [cmatthews@sentekconsulting.com](mailto:cmatthews@sentekconsulting.com).*