A STRATEGY PAPER FROM CENTER FOR DIGITAL GOVERNMENT

SERVICE-ORIENTED ARCHITECTURE:
Making Collaborative Government Work

A natural synergy for service-oriented governments committed to doing the public's business

Service-Oriented Architecture:
Making Collaborative Government Work

"Service is government's only business. SOA should be its architecture."

— **Paul W. Taylor,** Chief Strategy Officer, Center for Digital Government

Introduction

In 1918, the Spanish flu pandemic began inauspiciously. It caused a form of viral pneumonia that resulted in "extensive hemorrhaging of the lungs that could kill the perfectly fit within 48 hours or less." Before the pandemic had run its course, over 40 million people were dead in less than a single year.[1] It was a global disaster and the most deadly human plague of the 20th century.

The 21st century comes with its own challenges, including the threat of new pandemics. Fortunately, new technologies have brought with them the hope of a brighter future.

Although the ability of governments to effectively stem pandemics and other threats over time remains unproven, the capacity of government to detect, monitor and respond to events is at levels that would have been unimaginable just a few short years ago. Much of that capacity relies on digital technologies which began doing the heavy lifting of government one half century ago. This paper proposes a more optimistic view of the capability of government to more effectively serve the people through the adoption of a more fully evolved architecture — one that, as its name suggests, is service-oriented. To be sure, Service-oriented Architecture is well suited to the myriad of transactions between citizens and their government, but an integrated future holds much greater promise than that.

In a world of growing complexity and competing demands, government needs to go beyond a patchwork quilt of custom interfaces among systems that have nothing in common and develop a modern architecture that is strong, stable and comparatively simple. Enter Service-oriented Architecture (SOA). If SOA holds the promise for stopping future pandemics in their tracks, imagine what it can do for the huge volume of routine transactions between government and the citizens and businesses they serve.

Imagining Government Service Delivery:
A Future Where SOA Helps the Internet Keep its Transformational Promise

Imagine for a moment a world 100 years after the devastation of 1918. In 2018, due to the ongoing threat of global pandemics, information about illnesses and diseases is immediately transmitted by all government, health care, public safety and transportation authorities to Global Alerts, a consolidated real-time data stream of events affecting localities, states, regions, continents and the entire world. In that way, worldwide Departments of Disease Control and Prevention and local health, transportation and public safety authorities can receive real-time updates on potential exposures and other threats. Of course most health incidents are routine, therefore "system noise" has been counteracted by powerful stream filters that matured in 2010. Since that time the stream filters have been deployed to cull priority alerts from alerts of negligible consequence. Testing, treatment, quarantine and release teams can be immediately dispatched to help prevent outbreaks and avert crises.

Since the deployment of the Global Alerts Stream in 2015, serious viral or genetic infections have been transmitted to no more than 500 individuals worldwide in the last three years — a remarkable effort in coordination supported by the modern syndromic sense and response systems.[2]

What once was science fiction can now be the new reality of government service delivery in leading jurisdictions. The SOA experience points to the potential of transformation for those governments that join the journey toward a re-architected future.

Making the Case for SOA

The previous scenario describes a level of data integration and communication that is not only possible but likely — and necessary — if governments take advantage of the modern approaches and tools available. In doing so, government can build the future into information technology planning. Yet government faces twin challenges to achieve such a goal — infinite demand for services and finite resources to provide those services. The delivery of services at the scale needed by government requires information technology.

Government can no longer justify unnecessary duplication of infrastructure that could otherwise be shared. Nor can it afford to build tomorrow's stovepipe systems today. Governments confront some hard choices: continue to approach and service button-downed, hard-coded, hardwired IT systems of today — or seize new approaches and tools that are "agile," "nimble," "flexible," and "adaptive." These are the descriptors forward-leaning government leaders look for in their tool set as they strive to meet public expectations and build trust with citizens they help protect.

One promising approach that governments are beginning to imagine and act upon is called Service-oriented Architecture (SOA). A more detailed treatment of this topic will follow but briefly stated: SOA is an approach more widely adopted to date in the private sector; one that is improving collaboration between customers, suppliers and trading partners in a way that wrings inefficiencies and costs from business transactions. As the SOA concept moves away from machine room consciousness and moves toward mainstream consciousness via executive directors' offices, government is in an ideal circumstance to adopt the SOA approach.

As a framework, SOA provides patterns for design, development, deployment and management of IT services that support public services in ways that are network-aware, reusable and "available to other participants in the network as independent services that the participants access in a standardized way."[3]

This paper explains why service-oriented government is the right approach at the right time for government today to enable collaboration among government, industry, and the public. Service-oriented government also presents a way to increase output of services while keeping labor the same or at a decreased level to help handle the upcoming job migration as government workers move toward retirement. Finally, SOA and an implementation known as Web services is the approach that will help provide the blueprint and the tools to spark greater data integration that can lead to a renaissance of government modernization.

"The Flowers and the Bees" of Service-Oriented Architecture and Web Services

Service-oriented Architecture (SOA) will have a revolutionary impact on the field of information technology and by extension, any place IT touches. This space is one in which IT becomes more embedded each day in the drive toward ubiquity or universal "intelligence in everything." In this emerging environment, SOA also requires re-thinking about how software is designed and subsequently, where and how it is deployed.

To better grasp the concept of SOA, it is important to first understand that it is not a technology. Instead, it is a way of thinking about service (not technology) that becomes clear as it is broken down into components. For example, in government, "food assistance" for the poor is a service or program. Providing a hunting, fishing or driver's license to a citizen is also a service – in this case, a regulatory service. Typically a service has a consumer and a provider or producer. An organization choosing to adopt a SOA approach would promote using information technology to automate the consumption and provision of services over the Web. In fact, it is the Internet and the Web's very existence that has allowed SOA concepts to flourish and take hold. In the software world, a consumer of a service (in this case a Web Service) is often a software application. The provider or producer relies on another piece of software as its agent (the Web service), that the application uses to do a discrete piece of work.

The beauty of a Web service is like that of a flower: it can attract a multitude of sometimes "unintended suitors" — that is, software applications or consumers. Web services increase in value when reused and increase in value through the power of Extensible Mark-up Language (XML), which makes Web services readily available. For example, in 2004 the State of Utah re-designed its payment engine as a Web service based on a new architecture. Instead of the logic and process of making a payment for a business license or a motor vehicle license being a tightly integrated part of each new transaction-based application, this reusable component was made to operate as an independent service that could be used by multiple applications. This service reduced timelines for integration from months to one week. The relationship between a software application and a Web service is described in IT jargon as being "loosely coupled," as opposed to "tightly integrated."

XML is the New *Lingua Franca*

Unlike most humans under most circumstances, computer hardware and software do not handle ambiguity particularly well. For example, they do not do a very good job of looking at a series of pictures of a home for sale and picking which is the best representation or at least the most appealing representation of the property being sold.[4] For computers to talk to one another and understand what is being said, certain linguistic standards must be applied for the communication to yield the expected result. In this case the lingua franca of SOA and of Web services is XML.

To illustrate the point, it is useful to return to the services mentioned in the SOA description — namely, hunting and fishing licenses and food assistance for the poor. In a world before Web services, an application for any of those services would typically ask for the information for each service, causing duplicative data entry by the citizen and duplicative processing of that information by multiple applications. As cases or licenses come up for renewal, the pre-Web services application often does not recognize the user and so the data may be re-keyed each time a renewal occurs. Granted, this process is evolving without SOA to the point that better linking with databases allows some level of recognition and form filling capability to reduce data entry time. Yet SOA, through the use of Web services, provides an opportunity to move this process to a new level of efficiency and integration.

For example, Web services offer the ability to realize a long envisioned but still elusive citizen directory. At the citizen-facing level, an interface to a sharable directory would capture basic information about an individual who might use a government service in the future — information such as name, address, phone number and e-mail. At their discretion, citizens could set individualized preferences to pre-select specialized services to match their needs and lifestyle. For example, an individual could be notified about all licenses up for renewal over the next five years including occupational and professional licenses, hunting and fishing licenses, or renewal of motor vehicle tags or driver's licenses. As more applications become available online and existing services are updated, this individual could become a consumer of the citizen directory Web service. The data that existed with this particular service would not need to be re-collected nor permanently stored by multiple applications. This Web service could also allow for the implementation of a "single sign on" for citizens where their roles, needs and preferences are matched with appropriate access controls.

A citizen directory that is implemented with identity management can improve efficiency and address many security issues related to applications and the integrity of transactions. A payment gateway implemented as a Web service could also handle all credit card and Automated Clearing House (ACH) transactions for state and local governments and universities.

These examples demonstrate a primary reason that Web services are useful: instead of tight integration they provide "a very loose coupling between an application that uses the Web service and the Web service itself." This allows either piece to change without negatively affecting the other, "as long as the interface remains unchanged." This flexibility allows software to be built by assembling individual components into a complete application.[5]

These Web services do not need to be government created or owned but can be used at no cost or for a fee. This opens a range of possibilities for governments in deciding who is best positioned to provide a specific service, including contracting with private partners and non-governmental, civic organizations. Suppose a government has created a "mash-up" between the Google Maps Web service and a database of sex offenders. Because Google offered the map service at no charge, there would be no formal contractual relationship beyond a license agreement.

What happens when a vendor decides to charge for a service it has been providing for free? This is a minor issue when it involves a single Web service. But what if an aggressive government decides to consume hundreds of services as they become available? Clearly, a legal framework for consuming private sector Web services will be warranted in the near future, as these services will inevitably proliferate on the Web.

Definitions

XML

Wikipedia accurately describes XML as a World Wide Web Consortium (W3C) "recommended general-purpose markup language for creating special-purpose markup languages, capable of describing many different kinds of data." It permits the sharing of data "across different systems; particularly systems connected via the Internet…Programs can modify and validate documents in these languages without prior knowledge of their form."

Service-oriented Architecture

SOA is "an architectural style whose goal is to achieve loose coupling among interacting software agents. A service is a unit of work done by a service provider to achieve desired end results for a service consumer. Both provider and consumer are roles played by software agents (computer programs) on behalf of their owners."

Although the definition of SOA may apply to many services, a Service-Oriented architecture as described here is a design that allows for the coordination and use of a collection of Web services. Another way of looking at it is that SOAs provide a "unified service infrastructure, which is composed of several services applications."[6] Information technology architects believe that SOAs help organizations respond more rapidly and cost-effectively to changing conditions by promoting reuse of existing IT assets across an enterprise. It also allows government agencies to provide, if you will, plug-and-play extensions to their legacy data systems thus makings these islands of data accessible via the Web.[7]

Web Services

According to the W3C, a Web service is "a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically Web Services Description Language (WSDL)). Other systems interact with the Web service in a manner prescribed by its description using Simple Object Access Protocol (SOAP) messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."[8] Software applications written in various programming languages, running on various platforms can use Web services to exchange data over computer networks like the Internet. This interoperability is due to the use of open standards. W3C[9] and OASIS[10] are the primary committees responsible for the architecture and standards for Web services and e-business.

Service-Oriented Architecture is Here to Stay

The summary judgment of a critical mass of industry analysts suggests that SOA is a fundamental shift in software architecture and part of what some refer to as Web 2.0[11] — that is, the "next generation" Web. For example, Gartner estimates that by "2006, more than 60 percent of the $527 billion IT professional services market will be based on the utilization of Web services standards and technology. By 2008, SOA will provide the basis for 80 percent of development projects."[12] Governments have tremendous opportunity here and should do whatever is needed to capitalize on this trend.

Where to Use SOA in Government

Author and consultant Joe McKendrick explores a number of real world uses for SOA that are applicable to government. McKendrick's original list has been adapted, summarized and changed to better apply to government.

Data Management
Data management streamlines access and links data users to services they need to process data.

Workflow Management
Business Process Execution Language (BPEL) is an XML-based language that brings together the protocols for the heavy lifting of automating business processes with the subtleties of business interactions. BPEL extends Web services to transactions, including modeling steps that work goes through to be processed in an end-to-end transaction.

Service Consolidation
Service consolidation can link to and leverage existing in-house legacy systems, making them access via a single citizen or business portal. It also permits such uses as online bill pay.

Constituent Services Integration
Integration could create a series of loosely coupled Web services and provide a single interface for all types of constituent inquiries, feedback and complaints. Constituent services integration could also integrate workflow, alerts and data feeds to each agency for processing and follow-up.

Improving External Partner Relationships
Improve these relationships by permitting local governments, educational institutions, state agencies, courts, the legislature and nationwide public entities and trading partners/suppliers to create a shared library of Web services components.

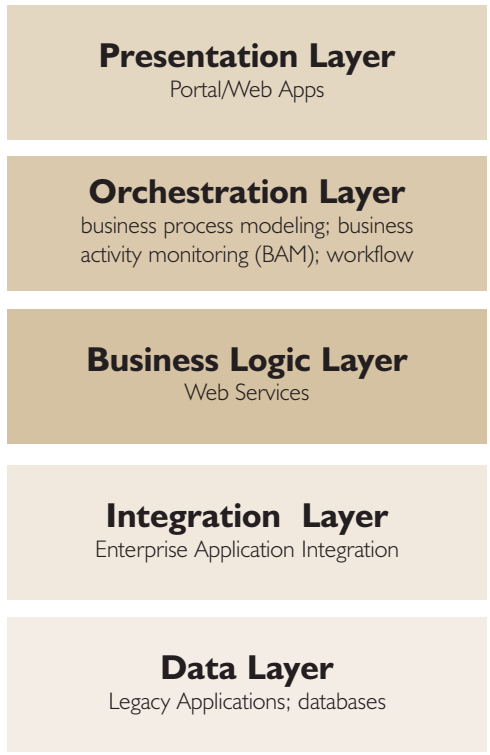
Creating Cost Efficiencies
Verizon Communications claims to have reduced IT costs by 50 percent by eliminating redundant systems inherited from the merger of Bell Atlantic and GTE. The SOA also helped integrate the operations of some 7,000 developers. Cost efficiencies of this type are also possible in government, particularly during IT organizational consolidations.

Reducing IT Automation Project Timelines
Reduce these timelines from months to weeks.

Application Consolidation, Re-use and Business/IT Alignment
To achieve consolidation, re-use and alignment, build a Web service once and allow multiple applications to use it.

Data Sharing Among Loosely Coupled Networks
Achieve data sharing among loosely coupled networks by expanding the access and sharing of data trapped inside hard-coded legacy systems.[13]

Service-Oriented Architecture



Some content for this sidebar was adapted from a presentation given by Toby Redshaw, CTO of Motorola, at SOA Executive Forum for Government, May 5, 2005.

http://weblog.infoworld.com/dickerson/001357.html

Web 2.0 – Simplicity First

SOA can be complex to implement or it can be easier to implement, depending on the approach. In essence, Web 2.0 advocates for simpler software — easier to build and use, cleaner and cheaper.

Microsoft Chief Technology Officer Ray Ozzie says it best in a recent memo: "Complexity kills. It sucks the life out of developers; it makes products difficult to plan, build and test; it introduces security challenges; and it causes end-user and administrator frustration. Moving forward…[we should]…explore and embrace techniques to reduce complexity."[14]

Developing a SOA and Web Services Mindset

Building the future of public service delivery means getting the architecture right. To that end, organizations need to begin developing SOA and Web services mindsets. Begin by just starting somewhere. Some state and local governments have done so by creating Really Simple Syndication (RSS) feeds for information that is of interest to the public and allowing an individual to pick or personalize the list of feeds that interest them. Using RSS is also a good place to start becoming familiar with the language of Web services, XML.

Another area that has become viable, although it has taken a number of hits since 2001, is using Web logs. There have been a fair number of legislators, public employees and at least one governor that tried blogging. Others have experimented with audio blogging or podcasts. Podcasting speeches or meetings are other emerging means of government-to-citizen communication. The beauty of podcasting is that it allows for portability — whether on a person's belt or in a vehicle during suburban commutes. The development of downloaded video podcasts is on the horizon. From these examples of RSS/XML deployments, governments can begin to create a Web services library that can be shared across states — from virtual travel services for state employees to citizen payment engines and citizen directories.

Starting with Data

Government data is everywhere, so a review of key government data is another good place to start building a SOA. Currently there is no single view, no uniformity, and virtually no reuse when it comes to data. Applications, as well as service-oriented applications, are about data. The vast majority of application development time today (some estimate up to 70 percent) may be spent finding and accessing disparate data. This is an indication that one of the first places to begin with SOA projects is by service enabling data.[15]

What "Done" Looks Like

The key to building a SOA is getting some quick wins and experience. Armed with the SOA mindset and a foundation of standards to build on, Web services efficiencies can be harvested. This begs the question: "What does 'done' look like?" As SOA evolves in the public and private sectors, "done" will be a fundamental shift in the architecture of software. Eventually Web applications will represent only a small portion of total usable code, the vast amount being consumed from loosely coupled Web services. This is a huge transition that will occur over the next three years. Of course, on the services side, "done" means a more agile, nimble, flexible, adaptive and service-oriented government.

On the Horizon – SOA Enablers

The new year opened with a breakthrough announcement that demonstrates the maturation of SOA through the release of a Service Component Architecture (SCA). SCA is a process and set of specifications that describe a model for building applications and systems using a Service-oriented Architecture. Service Infrastructure is a new type of software, necessary for extensive SOA adoption. Such standardization efforts free up developers' time. Instead of spending time on implementation and maintenance, developers can spend more valuable time developing business solutions.[16]

Action Plans: How One Government Got Started with SOA

Consistent with the state's world view — namely, "We relish competition and cherish our champions for their willingness to push beyond conventional boundaries to reach new heights of success," Kentucky has recently adopted SOA as the state's new platform for governing now and in the future.

According to Vibhas Chandrachood, executive director of Application Development for the Commonwealth of Kentucky, government is under increasing pressure to seek new IT solutions in the wake of baby boomer retirement. Chandrachood explains that it takes 150 people to manage these applications. "We currently have the personnel," says Chandrachood, "but we may not have that luxury in the future. So part of SOA is answering the question of how to do more with less."

Kentucky decided to implement SOA through the use of an Enterprise Service Bus (ESB (see next page for definition)) and is using SOA to extend and integrate legacy applications without relying on point-to-point interfaces. Chandrachood describes the old way of integration as "burning money" but uses more hopeful terms when he says that "data suggests that SOA is the most effective and cost efficient way to move forward."

Still, using an ESB to extend legacy systems while simultaneously building new applications in a

SOA environment has its challenges. Key among them, says Chandrachood, "is to understand and articulate what business problems SOA will solve." He also explains that developers need to carefully adhere to SOA architecture, framework, and to use open protocols like XML.

Kentucky is currently working on several applications to use the ESB and also building a core set of rules that run via that Bus. These services can be called upon by a variety of applications. One of the first systems identified by their business case process to come online with the ESB is the e-collections system of the Department of Revenue. Chandrahood explains that 150 agencies collect money and all have a need to talk to the tax system. This project will be followed by another heavily utilized database, the state's accounting system. In addition to these data integration efforts, the state is also building a Web service for address validation that Chandrahood describes as "a simple and reusable component of almost any legacy application. Build it once and it can be leveraged by anybody else," he says.

Action Steps for Moving Government Toward SOA

Service-Enable Data First
Look for the most heavily utilized databases with the most requested data elements and begin exposing this information first.

Utilize an Enterprise Service Bus
An ESB is software (a messaging engine) that makes services reusable and available to users, applications, business processes, and other services.

Identify Frequently Repeatable Processes
These will exist across agency boundaries and are candidates for re-use.

Establish Standards
Application developers need to understand, support and follow standards. Organizations need to provide developers with adequate training to successfully make the shift to utilizing Web services.

Just Start Experimenting
Develop SOA and Web services mindsets by experimenting with RSS (Really Simple Syndication) news feeds on agency Web sites. Have the RSS feed provide information that is of interest to the public or employees. Encourage employee Web logs, viewable at least from inside the firewall. Try audio and video podcasting for government-to-citizen communications.

Create a Web Services Library
Develop a library of Web services that can be shared across states. Provide a broad range of data, from virtual travel services for state employees to citizen payment engines and citizen directories.

Conclusion

Governments can no longer justify unnecessary duplication of infrastructure that could otherwise be shared. Nor can governments afford to build tomorrow's stovepipe systems today. Governments must confront some hard choices about the future. Will government allow itself to be characterized as the button-downed, hard-coded, hardwired IT systems of the past — or instead embrace the new approaches, methods and tools that a Service-oriented Architecture entails?

Governments and their leaders have the opportunity to rekindle a formidable and abiding trust among citizens. As trusted servants of the people, governments will receive the well-earned credit as they become more agile, nimble, flexible, adaptive and service-oriented governments.

References

[1] World Health Organization, "Avian influenza: assessing the pandemic threat." January 2005. http://www.who.int/csr/disease/influenza/H5N1-9reduit.pdf
[2] International Society for Disease Surveillance. http://www.syndromic.org/index.php
[3] Wikipedia, "Service-oriented Architecture." Updated January 17, 2006. http://en.wikipedia.org/wiki/Service_Oriented_Architecture
[4] amazon.com, "Amazon Mechanical Turk (Beta): Artificial Artificial Intelligence." http://www.amazon.com/gp/browse.html/ref=sc_fe_c_1_345361_3 103 53354064557436?%5Fencoding=UTF8&node=1587911 1 &no=3435361&me=A36L942TSJ2AJA
[5] World Wide Web Consortium (W3C), "Web Services Glossary: W3C Working Group Note." February 11, 2004 http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice
[6] Hao He, "What is Service-Oriented Architecture?" September 30, 2003. http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html http://inews.webopedia.com/TERM/S/Service_Oriented_Architecture.html
[7] Phil Laplante and Thomas Costello, "CIO Wisdom II: More Best Practices." p. 121, 2005.
[8] World Wide Web Consortium, op.cit., glossary.
[9] World Wide Web Consortium (W3C) http://www.w3.org
[10] Organization for the Advancement of Structured Information Standards (OASIS) http://www.oasis-open.org/
[11] Tim O'Reilly, "What is Web. 2.0?" September 30, 2005. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=5
[12] David W. Cearley, Jackie Fenn, Daryl C. Plummer, "Gartner's Positions on the Five Hottest IT Topics and Trends in 2005." Gartner, May 12, 2005 http://www.gartner.com/DisplayDocument ?doc_cd=125868
[13] Based on Joe McKendrick's *Ten Examples of SOA at work, right now*. January 2006. http://blogs.zdnet.com/service-oriented/?p=508
[14] Ray Ozzie, "Internet Services Disruption," Microsoft Memo, October 28, 2005, courtesy of Dave Winer and Scripting News. http://www.scripting.com/disruption/ozzie/TheInternetServicesDisruptio.htm
[15] BEA Systems, "BEA AquaLogic Data Services Platform," 2006. http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/aqualogic/data_services
[16] Robin Cover, "IT Vendors Promote Service Component Architecture (SCA,)" Cover Pages, OASIS, December 7, 2005. http://xml.coverpages.org/ni2005-12-07-a.html

100 Blue Ravine Road
Folsom, CA 95630
916.932.1300 phone
916.932.1470 fax
www.centerdigitalgov.com

Underwritten by:

BEA Systems, Inc. (NASDAQ: BEAS) is a world leader in enterprise infrastructure software, providing standards-based platforms to accelerate the secure flow of information and services. BEA product lines — WebLogic®, Tuxedo®, JRockit®, and the new AquaLogic™ family of Service Infrastructure — help customers reduce IT complexity and successfully deploy Service-Oriented Architectures to improve business agility and efficiency. With BEA platforms for building services and managing the entire SOA lifecycle, companies can confidently move their SOA projects from pilot to enterprise-wide production, even in heterogeneous environments.

Headquartered in Silicon Valley, BEA is a billion-dollar company with 15,000 customers worldwide served by 76 offices in 36 countries. For more information please visit **www.bea.com.**

CENTER FOR DIGITAL GOVERNMENT

Acknowledgments:

Paul W. Taylor, Ph.D., Chief Strategy Officer for the Center Digital Government and the Center for Digital Education
Al Sherwood, Senior Fellow for the Center for Digital Government and former deputy CIO for the state of Utah

About Center for Digital Government:

The Center for Digital Government, a division of e.Republic, Inc., is a national research and advisory institute on information technology policies and best practices in state and local government. Its advisory services, online resources and special reports provide public and private sector leaders with decision support, knowledge and opportunities to help effectively incorporate new technologies in the 21st century.

The Center's Strategy Papers provide two decades of experience and insight into the most critical IT topics governments are faced with today. The papers address important policy and management issues and offer strategic approaches for planning and implementing technology, funding sources and case studies from jurisdictions.