Introduction

Expectations of public institutions are rising. The arrival of the internet has led citizens to expect faster, better and more responsive access to government services. Simultaneously there is increasing pressure to provide more open and transparent policy implementation, including detailed performance monitoring and benchmarking. Finally, as the world gets more complex and unpredictable, public institutions need to respond quickly and flexibly to unforeseeable events. All these demands come at a time when public expenditure is more scrutinized than ever before.

Public institutions are seeking technology solutions to many of these challenges, and the move towards "e-government” is well underway. With the enthusiastic participation of IT vendors and systems integrators, many individual e-government initiatives have been spectacularly successful. But there have been many failures even more spectacular.

It has been argued that one of the causes of failure in e-government initiatives is the conflict between purchasers and private sector vendors and service providers. While a public institution shares many concerns with a private sector buyer, they have other completely different priorities, needs and duties including:

- **Longer time horizons** - planning in decades rather than quarters.
- **Equality of access** - the need to reach all citizens.
- **Respecting boundaries** - between departments and public and private sectors.
- **Serving the public interest** - including data protection and privacy issues.

E-government projects have many stakeholders, and may involve multiple independent agencies with different systems, skills and requirements. While these projects can present major social, political and technical challenges, it is the communication between agencies and systems that delivers the true benefits of e-government. It is therefore imperative that public institutions take the widest possible view of the many ways e-government can—or should—be implemented before committing to specific architectures, technologies or vendors.

This white paper discusses some of the principles that should be considered in any e-government project, and identifies specific causes of failure which should be avoided.

Only the Data Endures

E-government initiatives typically span multiple agencies, and at minimum are likely to involve several systems. No large distributed organisation can “join up” its IT without an integration strategy and an integration infrastructure. This infrastructure needs to be planned as carefully as any other infrastructural investment - think of roads and public utilities. It should be clearly thought out and should have a projected lifespan at least several decades long.

The integration strategy should last for decades

This can be difficult with IT where the life cycles of specific technologies are measured in years, not decades. Individual computers are obsolete in about 3 years. Back-end applications like databases have a lifespan of about 2-5 years before needing costly, complex upgrades. Programming languages and system architectures (dumb terminal, client/server, web browser) last about 10 years. Irrespective of lifespan, there is certain to be a broad variety of systems, languages and applications across agencies at any given point in time.
This makes it impossible for any long-term integration infrastructure to be based on a specific application, language or system architecture.

The only way to build an integration infrastructure to last is to base it on the one thing least likely to become obsolete - the data. No matter what happens to technology, people will be born. They will get birth certificates and passports and drivers licenses. They will work and pay taxes and receive healthcare. They may get married and have children. They will build houses and companies. They will, eventually, die. E-government is about making it easier to create, update and secure data so that public institutions can more effectively deliver the services that the citizens need throughout their lives.

The implication is clear. E-government Infrastructure should be based on the data that it needs to store and process. Applications and specific technology implementations should be considered expendable, and should in no way determine the integration infrastructure. The goal must be to ensure that when they are inevitably replaced with something faster, simpler or more powerful, the replacement causes minimal disruption to the network of systems that comprise e-government.

One rarely has choice in the way data is stored within individual applications but it is relatively immaterial to the overall e-government framework. The key is to ensure that the data can flow between systems and agencies in a format that can be easily understood by all.

Luckily there is widespread global consensus on the best way to exchange information - a format known as XML (eXtensible Markup Language). XML is the obvious choice because unlike other formats, humans can read it, all computer systems can be modified to understand it and it can contain not only raw data, but information about what the data means through embedded data-description tags.

Using XML representations of the data as the basis for e-government communications is therefore becoming a standard approach. We would go further and suggest that XML representations of processes, services and documents become the cornerstone of e-government integration strategy because XML is:

**Open**: it is not owned by any organization

**Transparent**: it can be read by any computer or person

**Responsive**: new tags can be added as necessary to describe new types of data

XML is therefore the ideal way to model not only the data, but the communication between the multitude of systems and agencies that participate in e-government.

**Minimize Technology Lock-in**

The recommendation to use XML comes with a large caveat - one needs to make sure that XML is being used in an open manner. There are ways to use XML and other open standards while perpetuating technology lock-in.

It is no secret that technology vendors seek to build technologies that make it difficult for their customers to switch to another vendor. Indeed technology lock-in is one of the most powerful competitive weapons of technology vendors.

It is impossible for purchasers to eliminate lock-in. One strategy that is gaining increasing momentum in government is to turn to Open Source software to reduce or even eliminate license costs, but that does not reduce the integration complexity for large-scale projects. There will always be switching costs to move from one technology to another. However buyers should understand and minimize lock-in because it quickly reduces flexibility and dramatically increases costs in any technology investment.
Vendors achieve technology lock-in in a number of different ways. Apart from inevitable product-specific training and skills investment with particular technologies, there are three key areas where public sector customers should identify and measure the degree of lock-in associated with any technology implementation.

The first and most obvious form is data lock-in. Most applications store data in proprietary internal formats which may or may not be easily accessible to third party applications. We have identified XML as the lingua franca of systems integration, and therefore at a minimum any system must be able to communicate and update internal data structures with other systems using XML. While almost all systems will support this to some extent, one should examine the data communicated via XML in an open format or whether it is an unusable proprietary representation simply wrapped in XML.

A second and more subtle area concerns the way open standards are supported. Software companies will invariably claim to support open standards, but will often do so in a manner that subverts the standards to secure lock-in. A classic tactic is to support an open standard, but to add proprietary extensions to the standard and deliberately blur the distinction between open and proprietary elements. A good example was seen during the browser wars. Microsoft and Netscape both supported HTML, but each added proprietary extensions to provide a “better user experience”. The result was that websites using proprietary features from one vendor did not function correctly with the competitive browser. Similarly, most databases include proprietary extensions to the open SQL standard, the use of which make it difficult or impossible to switch databases.

This tactic is particularly effective on commercial purchasers of technology who are invariably focused on immediate business performance. They gain a short-term competitive advantage by delivering a product or service based on an innovative but proprietary technology feature. Unfortunately they incur a much higher long-term cost because their new process is permanently dependent on a single supplier. The suppler can later extract revenues capped only by the business value of the new process through expensive consulting engagements and/or mandatory product upgrades.

It is worth noting that in the technology industry, proprietary new features offering compelling benefits quickly get replicated by competitors and eventually standardized. Once they are commoditized they can be adopted without fear of lock-in. In some cases the benefits offered by proprietary extensions may outweigh the costs of lock-in, but public institutions should take a longer-term view than their counterparts in private industry and should be more cautious of basing processes on proprietary features or extensions.

The most effective way a vendor can secure lock-in is to control the way that their product is used and accessed. Computer systems generally expose their features to software developers and other systems through an Application Programming Interface (API). The API can consist of hundreds of separate low-level commands which provide very basic functionality, and each is unique to that system. Developers who understand and can work with these APIs are invariably the most scarce and expensive.

By forcing customers to interact with their software though a proprietary API, vendors gain extraordinary lock-in because the buyer’s business processes are tied directly to the internal logic of the vendor’s system. Replacing the system involves a massive re-engineering of every other computer or process that interacts with the system.

To make matters worse, APIs tend to evolve over time with product upgrades and customers are often put on an endless treadmill of expensive system integration as individual back-end systems are updated.

Many technology vendors advocating XML and web services as a panacea for systems integration problems are simply placing their existing proprietary API functions in a new and glossy XML format. This does nothing to reduce vendor lock-in because external processes are still closely tied to the internal application logic through a slightly different channel. Even if Open Source software is used, integration costs remain high because of API-driven complexity.

The most effective way to minimize this form of lock-in is to keep different systems loosely coupled. Instead of using many low-level types of interaction with each system (i.e. the vendors’ APIs) one should develop a limited number of more powerful high-level ways that systems can interact with each other and with the overall e-government infrastructure.

The strategy is to carefully define high-level interactions as XML representations of real documents that flow between systems and agencies. Modeling the
integration strategy on real data and processes is orders of magnitude easier than modeling it on specific APIs, and breaks the direct dependence on fragile volatile interfaces.

This creates a clear level of abstraction between e-government - a permanently open, transparent and responsive set of processes - and the often closed and proprietary technologies that necessarily support it.

By making XML representations of the e-government business processes the core of the integration infrastructure, public sector buyers achieve a powerful new tool when procuring new systems - they can mandate compliance with detailed business-level data standards as part of the functional specification.

To summarize - most software business models are based on customer lock-in and the creation of islands of automation that are difficult to integrate into other systems. Vendors are sophisticated in the way that they implement open standards and protocols to lock-in customers, and a locked-in public sector customer is every vendor’s dream. Owning business-level data structures and standards is the key to avoiding lock-in. As a public sector customer, one cannot rely on vendor supplied check-sheets of supported standards - rather one needs to evaluate in a more critical and sophisticated manner the way the proposed technologies fit together and what it might cost to change or update individual components.

Separate Public and Private Concerns

One key difference between e-government initiatives and private sector projects is the separation of concerns on the project.

A private-sector project is intended to create a competitive advantage for its owner by offering a unique and compelling value to the user of the system. This often leads to architectures that are tightly coupled and require centralized control of the entire system from the user experience to the back-end technology. In the commercial world, this is perfectly acceptable because there will usually be multiple competing providers and the market is free to decide which of the offerings provides the best value.

With e-government projects, the key concern is to deliver universal access to government processes and services to the widest range of potential end users. In the medium term, it can be difficult to predict how, when and where those services will be used.

The most sensible approach therefore is not to seek end-to-end control of the entire system, but to separate the front-end from the back-end. Public agencies can develop back-end services and a core set of front-end interfaces, while allowing others to develop novel and value added services for citizens or indeed government itself.

A good example is US personal income tax returns, which can be difficult for even highly numerate people to calculate. Rather than attempting to create a standard spreadsheet or even an application to assist the preparation of filing, the IRS simply published the acceptable data format for tax returns. This spurred the creation of an entire industry which now supports many providers of tax filing software packages and online services at prices from zero to several hundred dollars.

No matter how much time and effort the IRS put into software development, it could not have matched the variety of novel filing modes offered, at no public cost, by the open market.

Equality of Access

In the private sector, much lip service is paid to the need to support multiple languages and interfaces accessible to people with disabilities, but commercial pressures limit the extent to which this actually happens. Public institutions quite rightly take their responsibilities in these areas much more seriously.

The fact of the matter is that no single front-end is capable of simultaneously serving both top-of-the-line graphical user interfaces and all the various disabilities that users may have. Therefore a clean separation of front-ends from back-ends is required to allow multiple specialized front-ends to all feed through to a single e-government process.

Privacy and Data Protection

Public institutions have a paramount responsibility to protect the confidentiality of the personal information entrusted to it by citizens. Citizens are rightly concerned that increasingly automated flows of information across government agencies and to the private sector creates the potential for misuse and harm. It is vital to establish and publish the principles and rules that will govern the use of private data across e-government, and Europe leads the world in data protection legislation.

Ensuring compliance with data protection laws and principles represents an extremely tough technical challenge for public sector organizations. It is vital to establish and publish the principles and rules that will govern the use of private data across e-government networks.

It is vital to establish and publish the principles and rules that will govern the use of private data across e-government networks.
challenge. Intervening in XML dataflows is done to remove explicitly private information as it flows beyond the boundaries of its home agency. This can remove many systematic breaches of privacy and data protection statutes, but the propriety of many data exchanges will depend on whether the subject of the data has consented to that specific exchange.

There is no simple and obvious balance between valid privacy concerns and the astronomical complexity of creating a technology-based enforcement of all relevant data protection legislation and principles.

One route is to ensure that all agencies and third parties contractually agree, under threat of appropriately severe penalty, to use information only in a manner consistent with current data protection laws and principles.

Partial and incremental automation is often the best approach to e-government

Fortunately e-government initiatives based on document-based XML workflows can readily log the "Who What and When" of workflow events across multiple agencies to provide a permanent audit trail, and the flow of information between agencies can be spot-checked for compliance. Suspected breaches of data protection can therefore be identified and investigated and appropriate action taken.

Incremental Automation

Public institutions have lifespans much longer than commercial enterprises, and their existing processes and procedures have evolved over much longer periods. Processes are likely to be more complex and are certain to be subject to more constraints than those in commercial enterprises.

When commercial enterprises engage in large technology infrastructure projects they often spend considerably more time and money than originally planned. It can be difficult to cancel out-of-control projects because none of the benefits are realized until every link in the chain is automated.

When public institutions attempt the same all-or-nothing approach to automating complex processes, the results can be truly catastrophic - billions of euros spent on projects that are ultimately cancelled with no productive results.

For a public institution, it makes sense to consider an incremental approach to e-government which phases automation over a period of time. One can identify the processes with the highest ROI or public utility and automate them first.

For example, in Ireland birth records are created by the General Register Office, but are used by many different agencies including health, taxation and education. Implementing an XML based publish and subscribe service enabled automatic notification of births to all interested government agencies and delivered an immediate and significant public saving by eliminating manual processing and human error. The exchange of other life-event notifications (for example deaths and marriages) can happen later as required. For many public sector business processes partial and incremental automation is the most appropriate route.

Unfortunately partial automation cannot be considered if the framework for integration is all-or-nothing real-time communication between chains of tightly-coupled systems.

Document-based workflows are the only approach to partial automation of processes and services. When System A wants to communicate with System B it sends a message over a reliable messaging network and forgets about the request until it receives a reply. System B services the request and replies when it has the time to do so.

The beauty of this approach is that no system expects a reply within milliseconds (which can never be guaranteed over large distributed networks anyway). It therefore does not know or care whether the request is being processed by another computer or a human being. One element of the process could easily be a public servant using a cheap and ultimately disposable browser interface to respond to requests which then re-enters automated workflow.

Considering e-government services as long-lived message-based conversations works because it is closely aligned to the way people and processes work in the
real world. It allows quick automation when possible and useful, while leaving other manual processes intact until there is a case to automate them.

In workflows where all processes are automated, the overall system has the appearance to a user of being completely real-time. It may be a few milliseconds slower than a tightly coupled API-based integration but for almost all e-government applications it is simpler, cheaper, more flexible and fast enough.

**Recognizing Departmental Boundaries**

There are many possible services in the public sector that are not available today because of interorganizational boundaries. For example a work permit application typically requires processes in foreign affairs, law enforcement, taxation and employment. A key benefit of e-government is the presentation of complex inter-agency processes to users as a single integrated service.

The primary barriers to enabling cross-agency processes and services are not technical - they are legal, political and procedural. Government agencies each have unique, long-standing and cherished heritages - with clearly defined physical, organizational and cultural boundaries.

Attempts to mandate a uniform technology strategy or instantaneous implementation from above will swiftly encounter resistance at departmental, regional and local levels. The notion of a single centralized e-government hub with everyone using identical XML data formats is extraordinarily difficult to achieve in reality.

Thankfully one does need to break down these long-established organisational structures in order to deliver e-government initiatives. The most effective approach is the one used for centuries by autonomous groups who have needed to work together for a common purpose - a federated structure. It is perfectly reasonable for different departments or regions to control their own technology environments and data structures while supporting shared processes and services across the wider network.

This can be accomplished by having multiple rings of data and process flow, linked by bridges that automatically convert XML data into the expected format as the messages flow between networks.

This approach enables the rapid creation of government-wide processes and services without the need for every affected agency to tear out and replace perfectly good systems. It also reduces the requirement for every single person in every agency to understand the bigger picture in order to deliver new services. As requirements for integrated processes and services at international level emerge, it is clear that only a federated information infrastructure will work.

**Policy Creation and Delivery Monitoring**

A critical function of e-government is the support it can offer to policy creation and delivery monitoring and improvement. Basing the architecture on the flow of service requests between agencies makes it possible to create holistic views of how collections of services, agencies and functions are working together.

Because these documents represent time-stamped checkpoints on real tax returns, passport applications and medical procedures, they contain invaluable information on service delivery. The way the documents pass through (or wait for) specific services and processes can provide an immediate and highly visual tool to identify service delivery bottlenecks, which can never be predicted in advance.

**A federated architecture enables inter-agency cooperation while respecting agency boundaries.**

The dataflows themselves should become the most effective real-time diagnostic tools to incrementally improve overall delivery at all levels of government. At the most senior levels, they can provide timely and relevant information for policymakers seeking ways to improve services and cut costs, and provide the best way for states to benchmark performance and delivery against each other.

**Flexible Reusable Services**

Within privacy and data protection constraints
e-government services should be offered widely and in a manner that makes them reusable by other agencies and third parties.

Natural disasters, diseases like Foot and Mouth and major one-off events all require the rapid creation and mobilization of cross-functional teams which may exist for short periods of time. A federated approach drawing on multiple reusable processes and services offers an invaluable mechanism for creating temporary agencies. The architecture should also allow the sharing of common services like identity management without having to centralize entire departmental systems - leaving ownership of the process or service where it belongs but removing duplication of effort across all agencies.

**The best way to deliver open, transparent and responsive government is through an open, transparent and responsive technology architecture.**

A wider point is that nobody knows what novel and useful services could in principle be developed under the umbrella of e-government - indeed it would be foolish to attempt to design a structure that placed limits on what could be possible. The creators of the internet understood this intuitively, and called it the end-to-end principle of network design. They rejected building sophisticated rules into the network and instead focused on getting any packet of data from one node to another in a reasonably efficient manner.

The success of the internet is due in large part to an architecture that allows the development and offering of new services without the need to seek permission from a central authority. The network itself should be relatively simple, and existing services on the network should have the potential to be combined in many ways to deliver new and useful services. Those who wish to see e-government fulfill its fullest transformational and integrative potential can draw much from the history and development of the internet.

**Conclusion**

In this paper we have identified the drivers of e-government initiatives and examined some of the technical, institutional and social barriers to successful implementation. Public institutions are long-lived entities and e-government in its most useful form interconnects multiple agencies. This necessitates a coherent long-term integration strategy and infrastructure.

By the time babies born today draw their pensions, most current leading technology vendors will be long forgotten. We do not know which ones. Of all the principles outlined the most important is that the technology must always be kept subservient to the data. If the data structures and dataflows are modelled correctly and owned by public institutions, the technology will take care of itself.

Throughout the paper we have argued that the best way to deliver open, transparent and responsive government is through an open, transparent and responsive technology architecture.

Flexible and reusable e-government services based on open data standards and technologies will provide the greatest and most enduring value to citizens, governments and the ideals of the democracies they serve.

**About Propylon**

Propylon offers products and services to build open, responsive and flexible e-government. It has implemented numerous successful e-government initiatives in Europe including parliamentary process management systems and inter-agency messaging services based on XML. Propylon defined the technical architecture and data structures for the Irish Public Services Broker which will provide integrated access to all e-government services for Irish citizens. Propylon’s technology is also used by companies including O2, KPMG, PricewaterhouseCooper and Thomson to provide highly secure and reliable services using XML.

More Information

To find out more about Propylon’s products and services for eGovernment please visit our website at [www.propylon.com/egov](http://www.propylon.com/egov) or contact Fergal Murray by phone on +353 1 492 7444 or via email at fergal.murray@propylon.com

© 2003 Propylon Ltd. This white paper may be reproduced and redistributed providing this notice remains intact. [www.propylon.com](http://www.propylon.com)