Making Social Security's Citizen Database Safe for the Future

By Jeff Gould, Peerstone Research

November 2009

OPEN MARKETS, OPEN SYSTEMS, OPEN NETWORKS

The Computer & Communications Industry Association

900 17th Street N.W.
Suite 1100
Washington, DC 20006
(202) 783-0070

WWW.CCIANET.ORG

Table of Contents

Making Social Security's Citizen Critical Database Safe for the Future

"Public information locked in a closed format has become increasingly unacceptable for governments."
- ODF Alliance[1]

Executive Summary

Under the weight of crumbling data centers and pressing calls for more open government, the debate surrounding Federal Information Technology modernization has recently leaped from the back office into the political arena. Current political priorities – from Health Care reform and deficit reduction to increasing transparency and accountability in Washington – are all inextricably linked to modernizing the Federal government's own IT infrastructure. Updating antiquated systems and making vital data more accessible will reduce costs and improve the ability of Federal agencies to deliver services to citizens. President Obama has made this topic a priority from the earliest days of his campaign. Discussing a bipartisan bill he sponsored last year with Senator McCain, then-Senator Obama highlighted the importance of increasing not only the transparency and accessibility of public data, but also its quality and usability. Doing so, he stated, would "tear down the barriers that separate the citizens from the democratic process and shine a brighter light on the inner workings of Washington."[2] Of all the agencies whose aging IT infrastructures require modernization, the Social Security Administration – with its great storehouse of citizen data – may be the one where the need is most urgent and yet most poorly understood by the general public.

Today the Information Technology used by the Social Security Administration is in crisis. At the heart of this system lies a vast database that touches the life of every American, from birth until death. Last year the Agency used this database to distribute benefits to 60 million recipients while recording the payroll contributions of 165 million tax payers. SSA also stores hundreds of millions of medical records, making it the largest repository of such records in the world. These records and the applications that use them are not merely "mission critical", they are truly "citizen critical". Unfortunately, the long-term integrity and accessibility of this precious information is in jeopardy, because it is locked into an obsolete mainframe database and applications architecture that cannot guarantee its safety and cannot make it available for the new uses which the Agency's evolving mission requires.

---

[1] ODF Alliance, "A Roadmap for Deploying Open Format Solutions in the Public Sector".

[2] Floor speech introducing Senate Bill S. 3077. June 3, 2008. [www.govtrack.us/congress/record.xpd?id=110-s20080603-37]

Through a combination of good intentions gone awry and inadequate planning for the future, SSA has sealed its treasure trove of citizen data into an outdated and proprietary data format that cannot be read by modern relational databases and cannot be processed by any kind of computer other than IBM mainframes. As reported by the National Research Council in its extensive review of the Agency's software infrastructure[3], SSA's existing primary database was created in the 1980s using a low-level machine language known as Assembler that runs solely on IBM mainframes. Over the years, the Agency's IT developers have surrounded this homegrown database (known as MADAM) with thousands of similarly low-level mainframe-only application programs written in Assembler or COBOL. These programs performed – and still perform – the critical work of processing the millions of items of citizen data that flow through SSA each day. Because MADAM does not understand the standard SQL data query language used by today's relational databases, the application programs that access it contain 10 to 100 times more code than would be required by a modern IT architecture. Furthermore, this code, much of which dates from the 1980s, uses languages that are rarely taught in the modern computer science curriculum and can only be understood by an ever-diminishing number of specialists (many of whom are themselves recipients of Social Security retirement checks).

At the urging of the National Research Council and other outside auditors, SSA has initiated a complex, multi-year program to replace MADAM with the mainframe version of IBM's DB2 relational database. However, information currently available to the public about this program suggests that SSA is implementing DB2 in a highly unusual and non-standard way that leaves intact many of the outdated mainframe-only application programs originally designed for use with MADAM. The crucial weakness of this strategy is that it fails to lessen the Agency's dependence on archaic mainframe-only database and application programming methods. This failing in turn threatens to make it far more difficult for the vast quantities of citizen data stored in this system to be accessed or used by more modern applications (even if these latter are themselves running on a mainframe). For example, the NRC points out that it will be a severe technical challenge to connect a database designed in this non-standard way to the Java-based programs the Agency hopes to use to expand its delivery of online services. Furthermore, in the event of an emergency, the non-standard data format used by SSA's current mainframe applications – which will apparently be perpetuated even after the migration to DB2 is complete – may make it difficult or impossible to process this citizen data on any type of computer other than an IBM mainframe, thus severely limiting the system's resiliency. The outmoded and growth-limiting character of SSA's mainframe software architecture parallels the similarly archaic status of the Agency's main physical data center. In both instances the Agency has failed to plan sufficiently far in advance for the replacement of

[3] Osterweil *et al.*, *Social Security Administration Electronic Service Provision: A Strategic Assessment*, National Research Council, August 2007.

obsolete infrastructures, with the result that the accessibility and security of vital citizen data have been put in jeopardy[4].

As part of the American Recovery and Reinvestment Act of 2009, Congress has provided $500 million to SSA for the construction of a new data center. We believe that Congress should also act to fund additional efforts needed to bring SSA's obsolete mainframe database and applications architecture up to modern standards. Although the Agency's outdated physical IT infrastructure has thus far attracted more attention from the public and Congress, its decaying mainframe software architecture poses an even greater danger to SSA's ability to carry out its mission. If the Agency fails to modernize its mainframe database and associated application programs now, the precious citizen data it manages on behalf of the nation could be locked into obsolete formats for decades to come. It would be as if the Agency had proposed to retain the 1980s formats of Microsoft Office for the long-term future storage of millions of vital citizen documents on the grounds that it was inconvenient to adopt newer formats. We cannot afford to store our vital national records in the electronic equivalent of ancient hieroglyphics whose meaning can only be gleaned by an ever-diminishing handful of experts. As described in the documents currently available to the public, SSA's strategy for migrating from MADAM to DB2 without changing its obsolete data formats is the equivalent of transferring a set of ancient hieroglyphics from a crumbling stone wall to a modern plate of stainless steel – such a strategy modernizes the substrate, but does nothing to modernize the archaic form of the text itself. With each passing year, the cost and technical challenges of creating a truly modern database for SSA's citizen information become greater, as does the risk of a catastrophic failure of access.

SSA's obsolete mainframe-only software architecture has many cascading consequences on the Agency's ability to fulfill its mission. One of the most damaging is that the architecture prevents the Agency from aggressively expanding its offer of online services to citizens, which the Obama administration has made into a top priority. Improving SSA's ability to deploy online services is particularly urgent, because the Agency's already overloaded field offices may not be able to cope with the surging demand generated by the boomer generation's retirement unless part of the workload can be channeled through the Internet. Until SSA embraces online services in the way that banks and other private sector financial service providers have, it is at risk of a catastrophic meltdown in its ability to process millions of citizen requests in a timely fashion. The non-standard format of SSA's mainframe database could also prevent it from making data fully accessible to its individual citizen owners, for example by allowing citizens to store their personal entitlements and benefits data with the financial service providers of their choice.

---

[4] For a review of the problems with SSA's physical data center infrastructure, see "Statement of Sylvester J. Schieber, Chairman Social Security Advisory Board To the Subcommittee on Social Security of the Committee on Ways and Means U.S. House of Representatives April 28, 2009".

SSA is locked into an expensive, inflexible technology strategy with a single major mainframe computer supplier. A store of citizen data as central to our national life as that of SSA should not be dependent on a single supplier. Such data should only be entrusted to IT architectures that respect open standards and guarantee long-term accessibility by relying on computer systems available from multiple competing sources. The dangers inherent in SSA's mainframe software have been documented at length by the National Research Council, the Congressional Research Service and the Social Security Advisory Board. Yet the urgent recommendations of these bodies for reform in SSA's IT strategy have gone largely unheeded. In order to lessen the long-term risk of a major software disaster involving citizen data, SSA must modernize its core software architecture at the same that it rebuilds the physical infrastructure of its data centers. Specifically, as SSA converts the 1980s era homegrown pre-relational MADAM database to the mainframe version of IBM's DB2 relational database, it should implement the recommendation of the NRC to adopt a standards-based model for citizen data using accepted best practices for relational database design and normalization.

As we have seen, the archaic character of SSA's mainframe software makes essential modifications such as the addition of web-based citizen access needlessly difficult to carry out. It also exposes the Agency to excessive costs. Because the incumbent mainframe supplier actively stifles competitors who could reduce costs as well as emerging technologies that could facilitate migration to alternative platforms, SSA (like other mainframe customers) pays far more for its large computer systems and their system-level software than it would if it used more modern software. As the NRC explains, the mainframe lock-in that the Agency suffers due to its legacy applications makes it "impossible for the SSA to take advantage of potentially lower cost alternatives" such as databases from Oracle or Teradata[5].

We agree with SSA management's position that a full migration away from the mainframe is not a feasible near-term option, and may not even be necessary in the long-term. However, we believe that the Agency, in view of its very considerable purchasing power, can afford to be more aggressive in its negotiating stance with its incumbent mainframe supplier. For example, SSA could buy additional flexibility for itself by insisting that this supplier permit market entry by providers of alternative mainframe solutions. Specifically, SSA should insist on its right to use innovative new technologies from IBM competitors that would allow it to run its mainframe applications more cost-effectively and that could help it to open up these applications to non-mainframe platforms in the future if it so chose[6]. The savings that SSA could

---

[5] *Social Security Administration Electronic Service Provision: A Strategic Assessment*, NRC, p. 63.

[6] Such new technologies include software such as Neon Enterprise Software's zPrime, which would allow SSA to run its mainframe applications more cost effectively on existing mainframe hardware using IBM's reduced price "specialty engines", and mainframe emulators such as those provided by T3T and TurboHercules, which would let the Agency's applications run together with IBM's z/OS mainframe operating system on less expensive industry standard

achieve from a competitive market for mainframe technologies would likely amount to billions of dollars over the lifetime of the supported applications. Even more importantly, they would enable SSA to free up development funds for the new online services that will be critical to its ability to fulfill its mission in the years to come.

The Social Security Administration, the Congress and the White House must act to insure the timely modernization of the Agency's "citizen critical" database and associated applications through the use of open standards and accepted industry best practices. This modernization must be conducted in a way that guarantees, in the words of President Obama, not only the transparency and accessibility, but also the quality and usability of SSA's citizen data for future generations.

In this White Paper we review the nature of the citizen information crisis at SSA and present a series of recommendations to address it, briefly summarized here:

First and foremost, SSA should carry out the modernization of its mainframe database and associated applications in a way that respects the principles of open data access. Above all, this modernization must remove all impediments caused by the use of mainframe technologies to a greatly expanded use of online services to citizens and to the provision of adequate Disaster Recovery facilities for its "citizen critical" data and applications. Congress must provide SSA with the additional resources necessary to accomplish this important transformation of its technical infrastructures. The SSA itself and its management should promise to pursue this goal with the greatest energy possible and without delay. The SSA must understand that this commitment is made not only to the President and to Congress but to the American people.

Second, Even if it chooses to retain for the long-term a database architecture based on the mainframe, SSA should be required to hold all critical citizen data in properly normalized data tables that can be read and modified by any of the widely used relational databases, including non-mainframe versions. Furthermore, SSA should separate all critical citizen data cleanly and completely from the programming logic of its applications. Although the continued use of COBOL may be acceptable with proper reengineering, the Agency should plan to replace all application programs coded in the low-level IBM Assembler language which is inextricably tied to mainframe hardware.

Third, In accordance with the recommendation of the National Research Council, citizens should have the right to consolidate their personal Social Security records with other financial and personal information in the computer systems of third party organizations of their choice such as banks, brokerage firms or retirement funds.

---

servers equipped with Intel or AMD microprocessors. The emulation technologies also lend themselves to "dual stack" architectures which, by running z/OS together with a modern server operating system such as Linux or Windows Server, can facilitate gradual migration of z/OS applications to these less expensive systems, as well as "offloading" some of the work from the high-cost IBM mainframe software stack.

Fourth, Acknowledging that it may not be feasible for SSA to migrate its core applications and citizen data off the mainframe, the Agency should nevertheless hold its computer system suppliers to a mandatory second source requirement. In particular, SSA should require IBM to resume its practice of technical and commercial cooperation under reasonable and non-discriminatory terms with suppliers of alternative mainframe technologies that are compatible with the IBM mainframe architecture.

Fifth, SSA should implement the SSAB's recommendation that it follow the IRS in confiding the assessment of its IT projects to a vendor-neutral third party. Like the IRS, it should insure transparency of its IT investment decisions by making public all votes of its IT Governance Committee and Executive Advisory Board.

Introduction: the challenge of citizen data at Social Security

No repository of citizen data in our nation is more important than that of the Social Security Administration. Social Security is the only government program that touches literally every person born in America as well as every employer operating in our economy. Since the program's inception in 1936 more than 433 million Social Security numbers have been issued and nearly $9 trillion in benefits have been paid. Last year 60 million American citizens and residents received $650 billion in retirement, survivor and disability benefits, while 165 million people paid Social Security payroll taxes on their wages. The Agency stores 250 million medical records – making it the largest repository of such records in the world – and will post over 270 million earnings items to workers' records this year. It will also issue nearly 250 million printed Social Security Statements[7].

Managing the information required to maintain our Social Security system in good working order is a monumental task. The health and well-being of every American depends upon the integrity and availability of this information. Insuring that this vast trove of citizen data remains safe and freely accessible in all circumstances, both now and in the future, ranks as one of the highest tasks and obligations of the United States Government.

Unfortunately, the information systems of the Social Security Administration, after decades of inadequate funding and oversight, now face a crisis of unprecedented proportions. The enormous volumes of precious and irreplaceable citizen data entrusted to SSA are today locked within obsolete software running on mainframe computer systems available exclusively from a single vendor. Although alternative mainframe technologies exist which are compatible with SSA's software and could significantly lessen its costs, it is not clear whether the incumbent vendor – IBM – will allow these competitors into the market. Furthermore, SSA's belated IT modernization program, initiated under pressure from outside Federal auditors and Congress, appears to have adopted a technical approach that, rather than opening up its IT applications and data to newer technologies, may actually perpetuate the Agency's dependency on obsolete mainframe software architectures.

Modern IT applications based on open standards structure data in such a way that the data can be processed and accessed on many different types of computers. This insures that access to vital information is not tied to the use of any one vendor's systems or proprietary standards. By contrast, the revamped applications at Social Security will still require the exclusive use of IBM-brand mainframe hardware and

[7] For a review of SSA's primary service responsibilities and workloads, see Kathleen Romig, *Social Security Administration: Workloads, Resources, and Service Delivery*, Congressional Research Service, February 2009.

software. Under SSA's current IT modernization plan, no other brand of equipment or software will be capable of handling the Agency's massive citizen database. Moreover, due to IBM's current policy of preventing alternative suppliers of IBM-compatible mainframe technology from entering the market, the Agency will have no option but to procure its mainframe hardware and software from a single supplier who is exempt from normal competitive market pressures. This supplier – IBM – will have only very limited incentives to engage in competitive bidding based on lower prices or superior innovation.

The crisis of SSA's obsolete mainframe database and associated applications comes at a critical turning point in the Agency's history, one marked by the onset of the Baby Boomers' departure from the work force. In the coming 20 years an estimated 80 million Boomers will file for retirement. Put differently, on each day that the Agency's Field Offices are open for business during the next two decades, an average of more than 16,000 new retirees will leave the workforce. Yet even today the Agency's 62,000 employees face overwhelming workloads that have led to serious and growing service delays. At the end of last year 750,000 benefit hearings cases were backlogged, while the average wait for a decision by an administrative law judge was over 500 days. Citizens now spend over 300,000 hours each month waiting in line to be served by overworked employees in SSA Field Offices.

Eliminating the backlogs and coping with the rising tide of demand requires a radical rethinking of Social Security's traditional ways of doing business. In the words of a letter addressed last December by the Social Security Advisory Board (SSAB) to President Obama's Transition Team,

- "the Agency has reached a tipping point wherein continuing to do business with outdated 20th century tools hinders its ability to meet the needs and expectations of the American public"[8].

Choosing the right modernization strategy at SSA is crucial, because the wrong strategy – that is, one that perpetuates the antiquated and proprietary architecture of the current mainframe-based software – will compromise the Agency's ability to expand use of the Internet as a delivery channel for its services. According to the in-depth review of SSA's information systems conducted at the Agency's request by the National Research Council in 2007[9], "online services offer the SSA an opportunity to improve its operational efficiency and to increase its total service capacity—in particular, to cope with its growing workload". The NRC recommended in its report that SSA make "an unambiguous, strategic commitment to electronic services as part of its long-term service delivery strategy". Yet the NRC also found that the Agency's obsolete mainframe data architecture was a serious obstacle to the deployment of online services, observing that "modernizing the underlying databases that support

[8] SSAB "Letter to the Transition Team for the President-Elect", December 16, 2008.

[9] Osterweil *et al.*, *Social Security Administration Electronic Service Provision: A Strategic Assessment*, National Research Council, August 2007.

the SSA's activities is critical to the Agency's effectiveness in the realm of electronic services".

The convergence of these two crises – the obsolescence of the Agency's proprietary mainframe applications and the surge in workloads due to the retirement of the Boomer generation – presents both a danger and an opportunity. The danger is clear: failure to transition the SSA's information systems to a more modern and effective architecture based on open standards could lead to a catastrophic decline in the Agency's ability to deliver services to millions of beneficiaries. At the same time, the recognition by SSA's management and outside auditors as well as by Congress that a crisis exists has created a once-in-a-generation opportunity to set the Agency's systems on a fundamentally new course. Only a standards-based IT architecture free from exclusive dependence on any one supplier's proprietary technology can meet the demands of an expanding workload, while respecting the administration's commitment to open government and information accessibility.

In the rest of this White Paper we first examine the scope of the citizen data access problem at SSA, then review the origins and causes of its current IT crisis, and conclude by suggesting remedies that can preserve the long-term integrity and accessibility of this most important of government information repositories.

Understanding the true scope of the citizen data access problem

In recent years, a worldwide movement has emerged in support of open standards for the storage and dissemination of citizen information on government computers and web sites. Until now, that movement has focused chiefly on the formats of office documents produced by PC-based productivity software such as word processors and spreadsheets. The goal of the movement has been to insure that all government documents are created and stored in universally accessible formats free from restrictions imposed by any particular software supplier. The introduction of internationally recognized open document standards such as ODF (Open Document Format), and their adoption by leading office software suppliers, marks a major step forward in the drive toward open government IT. For example, widely used office software suites such as Microsoft Office and OpenOffice now support ODF, which has become an official standard ratified by the International Standards Organization (ISO) in Geneva. Thanks to these developments, citizens around the world can be confident that the millions of PC-based office documents created by their governments every day will remain accessible far into the future, without regard for the particular technology used to create or edit them.

However, the adoption of open standards for PC document formats, although it represents a significant step forward, is not sufficient to guarantee the universal accessibility of all citizen data. Indeed, when we consider the whole vast scope of information stored and processed on government computers such as those of the Social Security Administration, we find that PC-generated office documents are no more than the proverbial tip of the iceberg. By far the greatest quantities of

information, and by far the most sensitive kinds of information from the point of view of citizens' rights and entitlements, are those that are created and stored for algorithmic manipulation by computer programs rather than for viewing and editing by human users. For convenience, we refer to these latter types of information as *programmatic information*, in contrast to the purely passive *documentary information* embodied in electronic office documents. The simplest way to explain this distinction is to observe that while documents are intended for direct use by humans, programmatic information in its native format can only be read by computer programs and must be transformed into documents – either on printed paper or computer screens – before humans can understand it. We will briefly elaborate on the importance of programmatic information in the following section, and then examine its implications for the openness and accessibility of the citizen data held in the SSA's computer systems.

How SSA's outdated IT locks in its citizen data

Perhaps the best-known example of programmatic information comes from the financial industry. Banks and other financial institutions record the amounts of money and other assets they hold for customers in special computer records called "accounts", which partially duplicate the structure of the paper-based account books of the pre-computer era. The amount of money in a given account naturally varies over time as the customer executes withdrawals, payments, deposits and other operations. These transactions are of course no longer recorded by pencil-wielding clerks, but by computer programs which perform the record-keeping work many thousands of times more efficiently than human employees. Indeed, the modern banking system, which encompasses hundreds of millions of customers performing billions of transactions per year, would not be possible without computers. The magnitude of the recent crisis of the global financial system simply underscores the extent to which civilization as we know it today depends upon such programmatic information, which is fundamentally different in nature than the information stored in office documents.

The information representing customer assets and transactions in the computers of financial institutions is not organized in human-readable document formats such as ODF. Instead, it is stored in intricately structured lists or tables that have been painstakingly crafted to make them easy to process by the specific computer programs in use at each financial institution. These programs contain the business rules (also known as "business logic") applied by banks to each customer transaction. The rules include both elementary arithmetic and higher-order logic such as the computation of interest, overdraft limits, foreign currency conversions, and the like.

In the 1960s and 1970s, the only computers available for serious business applications were the large, costly systems known as mainframes, whose hardware and software were entirely proprietary to their original manufacturers[10]. Applications developed for these mainframes by banks, corporations and government agencies of this era stored their programmatic information in special disk file systems. These were essentially indexed lists of records, where each record corresponded roughly to a line in an account book. The custom mainframe programs designed to access this information had to scan through long lists of records using complicated logic that was difficult to develop and maintain. Such programs were most often written in COBOL, a programming language first introduced in 1959 and regarded as the oldest computer language still in common use. Although COBOL is usually associated with mainframes, it can also run on modern industry-standard servers equipped with Intel or AMD microprocessors and operating systems such as Linux, Unix or Windows Server. In addition to COBOL, many large mainframe users have also developed programs in Assembler language, a low-level machine language closely tied to the specific hardware architecture of the mainframe that cannot function on other computers.

Mainframe computers were designed to process large amounts of mission-critical data created in large corporations and government agencies. This data is what we have referred to as programmatic information, because it must be organized in such a way that application programs can easily access it and manipulate it. From the very beginning of the mainframe era, the question of finding the right software tools to organize and store such data was crucial. As we have seen, early mainframes stored their data in what were essentially glorified file systems, not unlike the folders and subfolders familiar to users of modern PCs. However, the need for more sophisticated and efficient data access methods soon became apparent. The 1960s and 1970s saw the emergence of the first true database systems, such as IBM's IMS (Information Management System), originally developed for NASA's Apollo moon program. These early databases, which were highly complex software "engines" that undertook the task of storing data and making it available to user applications as

[10] The early mainframe market was highly competitive, with a number of manufacturers such as IBM, Sperry, Burroughs, NCR, General Electric and Honeywell all offering their own competing mainframe designs, each with its own proprietary hardware, operating system, middleware and database. Customers benefited from the resulting competition in mainframe pricing and technical innovation, but were frustrated by the near-impossibility of shifting complex and expensive custom applications between different mainframe architectures. However, in the mid-1970s, under pressure from regulators, IBM began offering its mainframe operating systems for use on non-IBM mainframe hardware made by the so-called "plug compatible mainframe" (PCM) vendors such as Amdahl and Hitachi. As a result this strategic effort by IBM to establish its own mainframe operating systems as the de facto standard, the mainframe market coalesced around IBM's hardware architecture. By the mid-1980s IBM compatible mainframes constituted the vast majority of the market, and the non-compatible mainframe manufacturers together with their non-IBM mainframe operating systems were marginalized.

efficiently as possible, offered good performance, but they proved difficult to program and manage.

In the early 1980s a new method for storing programmatic information was introduced known as the relational database, which soon became the de facto industry standard for both mainframe and non-mainframe databases. Relational databases store their information in linked tables that are designed to allow fast access while minimizing the risk of errors or corrupted data. The data tables can be thought of as similar to spreadsheets, except that a typical business application will, for reasons of efficiency, divide its data into dozens, hundreds or even thousands of interlinked tables. Relational databases are usually employed together with helper software known as "middleware" that allows large numbers of users to access the database simultaneously from remote terminals, PCs or web browsers. The business rules used to manipulate the data stored in relational databases can be written in almost any programming language, including COBOL, Java, C++, C#, or PHP. But when these programs need to read or write data in the database, they use a different, more specialized data access language known as SQL (Structured Query Language), which is an international standard recognized by ANSI and ISO. All of the leading relational database products on the market today are based on SQL, including those from IBM, Oracle, Microsoft, and Sybase, as well as open source alternatives such as MySQL and PostgreSQL.

Although relational databases based on the SQL standard have been in widespread use for at least a quarter of a century, SSA's core information systems today still use a pre-relational data architecture based on 1970s era technology. Because this antiquated technology runs only on IBM mainframes, the result is that the Agency's most vital stores of citizen data remain completely inaccessible to any other kind of computer system. Before looking at SSA systems in more detail, we will briefly contrast this state of affairs with the typical architecture of modern business applications.

What modern IT applications should look like

Modern business computer applications deployed in corporations and government agencies almost always consist of three main pieces:

- a relational database, which is an off-the-shelf product supplied by a software vendor;
- middleware, which is also an off-the-shelf product (often known as a "transaction monitor" or "application server");
- the business logic, which is written in a programming language such as COBOL or Java, and which can be a one-of-a-kind program specifically developed by (or for) the user organization, or may in some cases be an off-the-shelf product (for example, an Enterprise Resource Planning package such as SAP).

In modern business applications, all three of these pieces can and usually do reside on different computers, which may use different operating systems and different microprocessors. Furthermore, while the business logic in an application built on this modern three-tiered architecture may be unique to the company or Government agency that owns it, the spread of open IT standards guarantees that components such as databases and middleware as well as the underlying computer hardware can be purchased from many different vendors. For example, it is perfectly possible – and indeed quite common – for a business application written in Java to run on a Windows Server computer while accessing data stored in an Oracle database on a Unix server.

If business circumstances dictate, a customer can readily replace some or all of the components in the example just given with equivalent products from different IT vendors, all without losing valuable business data or business logic. For example, the customer's Java business application could migrate from Windows Server to a Linux server, while the Oracle database running on Unix could be replaced by a Microsoft SQL Server running on Windows Server. To take an entirely different example, consider the case of a customer using a business application written in COBOL that accesses data stored in a DB2 database on an IBM mainframe. If the data tables in the database are properly structured according to modern standards, and if the COBOL program does not use proprietary features of the mainframe environment, then both components can migrate to Unix servers supplied by IBM or another vendor, thus allowing the customer to benefit from very significant cost savings[11].

[11] It is important to point out that this ideal scenario of standards-based application portability between mainframe and server environments unfortunately does not apply to the vast majority of existing mainframe installations, due to the frequently outdated and non-standard architecture of their applications and associated databases and middleware. In particular, it is not a plausible model for the near-term evolution of SSA's IT systems. It is true that a small percentage of mainframe users have successfully "rehosted" relatively simple and self-contained mainframe COBOL programs onto modern non-mainframe servers, albeit at

Today there are many such multi-vendor "mix and match" deployments in Federal and local agencies as well as in the corporate sector, all made possible by IT vendors' adherence to standards such as the TCP/IP Internet protocol for networks or the SQL query language for relational databases. In a market defined by standards, IT suppliers must compete on the quality and functionality of their products rather than locking customers into proprietary solutions.

In addition to shared standards, modern business applications also benefit from shared principles of good application design, commonly known as "best practices". These include the division of large programs into smaller and more manageable chunks or "modules", the separation of the "business logic" of programs from the underlying format of business data, and the careful structuring of such data according to standard principles known as "database normalization".

Applications that fail to implement modern IT standards and shared best practices suffer from severe limitations in the accessibility and security of their data as well as high costs. In particular, such applications are:

- more costly to develop and deploy, because they typically rely on proprietary hardware and software supplied by an incumbent vendor who is protected from competition;
- more costly to maintain and adapt, because they require less commonly available skills linked to non-standard technologies;
- vulnerable to unexpected failures and declining performance, due to their use of obsolete architectures that can often neither be upgraded nor easily modified;
- unable to offer adequate guarantees that the information they create and store will be accessible in all times and circumstances;

---

considerable cost. However, for most mainframe users, the sheer complexity of their applications and the intricate association of these programs with the underlying mainframe operating system, middleware and database rule out simple rehosting strategies. For example, the mainframe applications portfolio at SSA contains many millions of lines of COBOL and Assembler code, which are in turn divided into thousands of individual programs that make calls to a non-standard mainframe-only database through non-standard mainframe-only middleware (itself written in low-level IBM Assembler). Many of these programs and database calls may be poorly documented or difficult for programmers trained in modern programming techniques to understand. Many of the mainframe operating system functions that they use have no direct equivalent in modern operating systems such as Linux, Unix or Windows Server (LUW). Consequently, a massive all-at-once migration of a complex mainframe application portfolio such as SSA's to a modern server hardware and operating system platform is not a practical possibility. However, a thorough modernization of SSA's mainframe applications and database, one that brings them into compliance with modern database and programming standards and architectures, can pave the way in the long-term to a controlled and incremental diversification of SSA's mission-critical hardware and operating systems.

Why SSA's information systems are in crisis

The origins of today's crisis in SSA's information systems date back to the early 1980s, when the Agency decided to develop its own custom database system, called MADAM (Master Data Access Method). This system, which is still the primary database in use today for SSA's core applications, was created using IBM mainframe Assembler language and a proprietary IBM indexed file format known as VSAM (Virtual Storage Access Method). According to the National Research Council, MADAM was believed as of 2007 to contain at least 30 Terabytes of data, representing essentially everything SSA knows about the hundreds of millions of individuals and millions of employers who are its contributors and beneficiaries. The business logic of SSA's core applications is embedded in a large number of IBM mainframe Assembler and COBOL programs. These latter are closely tailored to the intricate and highly non-standard format of the data in MADAM.

In the early 1980s mainframes were the only type of computer capable of meeting the requirements of a user as large as SSA. But the Agency's decision to build its own database rather than purchase an off-the-shelf product such as IBM's DB2 relational database (released in 1983) was not a foregone conclusion. Indeed, in a 1986 report the Congressional Office of Technology Assessment was severely critical of SSA's choice:

- "The most controversial accomplishment of the data integration effort is perhaps the Master Data Access Method, or MADAM, the file management system that SSA developed when it converted from tape to disk storage. Many experts thought that SSA should have sought or adopted off-the-shelf software for this purpose which would be maintained by vendors, rather than developing its own which it must maintain (that is, improve, modify, and update)… MADAM is apparently a very complicated and poorly documented system, so that only a small group of people are sufficiently knowledgeable to operate it, yet it is the basis of the SSA's data management. This constitutes a particular vulnerability to smooth operations if there is any short-term emergency, sudden work force reduction, or drastic reorganization."[12]

More than two decades later, the NRC concluded in its 2007 report that MADAM is "technologically obsolete and functionally primitive compared with readily available commercial technologies and products". The NRC went on to state that "because such outdated technology is used for the SSA's master database, it raises barriers in terms of the feasibility of linking it to new Internet applications" and therefore constitutes "a significant barrier to the SSA goal of providing improved and expanded

[12] U.S. Congress, Office of Technology Assessment, *The Social Security Administration and Information Technology*, October 1986.

electronic services to users". The NRC further notes that SSA's mainframe database technology is "unavailable for 24/7 access" because of its reliance on 1980s era batch update methods.

In addition to the home-grown MADAM system, SSA uses a variety of more modern relational databases for administrative applications and management information. The providers of these relational databases (all of which use SQL) include IBM, Oracle and Microsoft. However, SSA's core "mission critical" applications all use the non-relational, non-standard MADAM system. These applications include those the Agency uses to administer its most vital services to American citizens:

- Old Age and Survivors Insurance (OASI);
- Disability Insurance (DI);
- Supplemental Security Income (SSI);
- Recording of employee earnings;
- Issuance and verification of Social Security numbers (SSNs).

Both MADAM and the Assembler and COBOL programs that implement SSA's core business logic run exclusively on IBM mainframe hardware and software. The hundreds of millions of citizen data records stored in MADAM cannot be directly used or understood by programs running on any other kind of computer. In effect, SSA and its database are hostages to the dictates of its incumbent mainframe supplier, which can set whatever prices it likes and control or limit the pace of technological innovation as it sees fit. But a store of citizen data as central to our national life as that of SSA should not be dependent on the good intentions or continued health of any single company, even if the behavior of that company is entirely above reproach. Such data should only be entrusted to IT architectures that respect open standards and rely on computer systems available from many competing sources.

SSA's IT modernization plan fails to guarantee data accessibility

There are ample signs that SSA is well aware of the need to address the crisis in its mainframe software systems. In its *Information Technology Vision 2009 – 2014* report, the Agency recognizes the strategic imperative of modernizing its mainframe database and the legacy applications that use it. The agency has also hired Booz Allen Hamilton to develop an enterprise database strategy. We do not question the seriousness of SSA's desire to address the problems in its strategic IT infrastructure. However, we are concerned that the Agency has not provided sufficient details of its plans in this area for proper public scrutiny and debate. To our knowledge, neither the *Information Technology Vision 2009-2014* report nor the Booz Allen study have been made available to the public. Furthermore, the Agency's own Office of the Inspector General, in a written response last June to a query from Senators Baucus and Grassley regarding SSA's strategic IT planning process, found that the Agency's planning documents "generally did not provide a detailed description of how the

Agency intends to address its future IT needs"[13]. The Social Security Advisory Board has also urged the Agency to engage in more detailed long-range IT planning, particularly with regard to its ability to deploy the IT infrastructure required to deliver services to citizens in the 2020 time horizon[14].

As far as we can determine from the publicly available documents, SSA still cannot say exactly when MADAM will be finally withdrawn from service. Nevertheless, after criticism from the NRC, the SSAB and other government auditors such as the GAO and the Congressional Research Service[15], the Agency has embarked upon a long-term modernization project that involves the gradual replacement of MADAM by the mainframe version of IBM's DB2 relational database. According to the SSAB, this project is not expected to be finished until some unspecified time "beyond 2014" and in fact may take a decade or more to complete. It appears that as of this writing two of the four major data files in MADAM have been converted to DB2[16]. The success of this conversion and the operational status of the target DB2 installation cannot be determined on the basis of presently available information. Unfortunately, according to the previously published findings of the NRC, it appears that the technical approach taken by SSA to the MADAM modernization project fails to meet fundamental requirements for a vendor-independent, standards-based IT architecture and fails to guarantee the universal accessibility of the citizen information entrusted to the Agency.

A basic weakness of the existing MADAM database and the core SSA applications that use it is that they are written in older programming languages – COBOL and mainframe Assembler – that lock them into IBM mainframe hardware. We have already mentioned the status of COBOL, which the SSAB perhaps somewhat unfairly characterizes as "archaic". However, the other language in widespread use at SSA, IBM mainframe Assembler language, which predates COBOL in both use and concept, undeniably is archaic, and is even more inextricably tied to the IBM mainframe hardware architecture. COBOL was designed in the 1950s as a high-level business programming language that theoretically could run on different types of computer hardware with the help of a translation program known as a "compiler". Over time COBOL's hardware independence was weakened by its close association in actual programming practice with proprietary mainframe-only middleware and

---

[13] SSA Office of the Inspector General, *Congressional Response Report: The Social Security Administration's Information Technology Strategic Planning*, June 29, 2009.

[14] Social Security Advisory Board, *Bridging The Gap: Improving SSA's Public Service Through Technology*, April 2009.

[15] In addition to the reports cited above from the National Research Council, the Office of Technology Assessment and the Congressional Research Service, see Government Accountability Office, *SSA Has Taken Key Steps for Managing Its Investments, but Needs to Strengthen Oversight and Fully Define Policies and Procedures*, September 2009.

[16] According to statements made at the meeting of SSA's Future Systems Technology Advisory Panel held in Washington DC on November 5, 2009.

operating systems. Mainframe Assembler, by comparison, is a low-level machine language that was never intended even in principle to be hardware independent. It runs solely on systems that implement the IBM mainframe instruction set, either in actual silicon (in the case of IBM's own mainframes) or in software (in the case of mainframe emulators we discuss later in this paper). Programmers versed in Assembler are extremely rare in today's IT labor market, and hence very difficult and costly to hire. Indeed, use of Assembler has become so unusual that, according to computer book publisher O'Reilly Media, it accounted for barely one quarter of one percent of all programming books sold last year. This is in stark contrast to the top five languages – C#, Java, PHP, JavaScript and C++/C – which collectively represented 56% of book sales[17].

Regrettably, SSA's proposed modernization plan does not appear to remedy the Agency's reliance on mainframe-only programming methods. Indeed, Social Security Commissioner Michael Astrue recently told the Agency's Future Systems Technology Advisory Panel that he "[did] not think that the Agency [would] be able to move away from COBOL completely"[18]. The National Research Council's 2007 report provides more detail on SSA's database modernization plan. It states that SSA intends to replace MADAM with the mainframe-only version of IBM DB2, while leaving intact all or most of the mainframe Assembler and COBOL programs that access the underlying data. Such an approach would eliminate the need for the Agency to develop new versions of its core applications in a modern programming language such as Java or C#. However, a very damaging consequence of this plan, if implemented, is that our Social Security system will still be using 1980s era computer code originally written for 1970s era IBM mainframes well into the middle of the 21st century – and perhaps beyond.

The Agency and its consultants (whose expertise according to the NRC is limited to IBM mainframes) may have chosen this approach because they believe it is easier to implement. But the decision to retain obsolete mainframe application code while attempting to migrate to a relational database like DB2 carries hidden risks and costs. In particular, this approach requires a highly unusual and non-standard use of DB2 in order to replicate the obsolete data formats of the old MADAM database. When relational databases are used according to widely accepted best practices, information is stored in a way that respects certain principles known as "normalization". Database normalization has many important advantages. It reduces the risk of errors when information is updated, and it makes it easier to design new applications to query the database that may not have been anticipated when the original applications were developed. Normalized databases represent individuals and events in ways that are easier for human programmers to understand, and they are therefore easier to maintain and modify. One key example of the kind of new application whose use is greatly facilitated by a properly normalized database is the

[17] O'Reilly Radar, "State of the Computer Book Market 2008, part 4 -- The Languages" (radar.oreilly.com/2009/02/state-of-the-computer-book-mar-22.html), accessed May 4, 2009.

[18] FSTAP Meeting Minutes, October 2008.

modern Java-based middleware the Agency needs to deploy in order to significantly expand its online services.

But it appears that SSA, in order to preserve the decades-old business logic embedded in its IBM mainframe Assembler language programs, has chosen to resort to non-normalized data tables in its DB2 implementation. In a disturbing but revealing example cited by the NRC, the Agency apparently plans to use *46 distinct DB2 tables* to represent the information concerning each *individual* beneficiary in the Supplemental Security Income Records database. Such a profligate data architecture is likely to prove very difficult to maintain in the future, particularly decades from now when the original designers have long since departed from the scene. Furthermore, it will be extremely challenging, if not impossible, to develop *new* applications that can make use of data structured in such highly idiosyncratic fashion. The Agency's ability to meet new needs and unexpected challenges in the future may thus be compromised. In the words of the NRC,

- "Such an approach is almost certainly going to experience poor performance, suffer from numerous potential update anomalies, and almost inevitably be the source of numerous long-term complications if actually adopted. The proposal of such a design in the first place is very surprising and illustrates the inadvisability of adopting a MADAM-to-DB2 conversion approach that does not include a rewrite of the actual applications to make DB2 calls directly."

Another serious drawback of the Agency's plan is that even though it involves replacement of MADAM by a modern relational database, it largely defeats the purpose of moving to an architecture based on open standards, because the resulting applications will remain tied to a single vendor's mainframe hardware and software. SSA's Assembler and COBOL programs will still only run on IBM mainframes, and it will be difficult or impossible for non-mainframe applications to access the non-standard data tables the Agency intends to deploy on its mainframe DB2 database. The NRC observes that the Agency's approach makes it "impossible for the SSA to take advantage of potentially lower cost alternatives" such as relational databases from Oracle or Microsoft or indeed the non-mainframe version of IBM's own DB2.

To understand the true magnitude of the dangers posed by the proposed single vendor IT architecture at SSA, it is useful to contrast the Agency's future mainframe data plan with the adoption of open document standards in the world of PC-based office computing. Prior to the development of XML-based standards such as ODF, some users of Microsoft's widely distributed Office suite worried that the word processor, spreadsheet and presentation documents created with that suite could not be readily accessed by other brands of software. Concern arose in government agencies in particular that documents stored in Microsoft's so-called binary file formats, which are proprietary to that vendor, were not universally accessible, and that their long term availability could not be guaranteed because it was subject to the continued viability and good will of Microsoft.

Now that both Microsoft Office and its leading rival OpenOffice implement ODF as well as Microsoft's own OOXML format, these very legitimate concerns have largely been allayed. Government and private sector organizations can now be confident that office documents they create in these standard formats will remain fully and freely accessible in the future, no matter which brand of software they choose to adopt. In sharp contrast to this embrace of open standards by all the major PC-based office software suppliers, SSA's planned retention of IBM mainframe Assembler and COBOL programs in combination with the mainframe-only version of DB2 is a huge step backward. It is as if the Agency had decided to retain the 1980s version of Microsoft Office for the long-term future storage of millions of vital citizen documents on the grounds that it was inconvenient to adopt newer formats.

Spending billions of taxpayer dollars for a mainframe modernization plan that will lock America's most important citizen data assets into a single vendor's proprietary computer systems for the next half century or more is simply not acceptable. SSA's mainframe software modernization plan – at least the version of the plan that has thus far been communicated to the public – is unnecessary, unjustifiably expensive and fraught with technical risk. We believe that it should be fundamentally revised. The only acceptable long-term solution for SSA's future information systems is one based on open standards, unhindered market competition for both the underlying hardware and operating system platforms, and a guarantee that the information entrusted to the Agency by its hundreds of millions of citizen-stakeholders will never be held hostage to a single vendor's technology.

SSA can benefit now from alternative mainframe technologies

A thorough-going modernization of SSA's software architecture is essential if these critical applications are to be freed from their dependence on obsolete mainframe programming methods. Although a complete migration away from the mainframe is not feasible at this time because of costs, risk and uncertainty, it is nonetheless essential for the Agency's next generation of information systems to be freed from their exclusive reliance on a single vendor's brand of mainframe computer hardware. However, these bedrock requirements for a truly open information system at SSA need not require the Agency to abandon all use of COBOL or IBM mainframes in the near-term. A diversification of the hardware and operating system platforms in use at the Agency for its most "citizen critical" applications is a sensible long-term goal. But reasonable observers will acknowledge that any long-term modernization plan must take care to avoid unnecessary strain on the Agency's organization and budget, and should not incur undue technical risks. Fortunately, good technical compromises are available that would permit the Agency to retain its mainframe hardware architecture and some of its mainframe programming code for a number of years while it performs a controlled and incremental transition to more a more diverse set of basic platforms. Indeed, with a modernized data architecture founded on relational databases and a suitably normalized data model, there may well be an appropriate role for

reengineered COBOL programs and mainframe-compatible hardware in SSA's IT infrastructure into the indefinite future.

With regard to the continued use of IBM mainframe hardware at SSA, several solutions exist that allow legacy mainframe programs (including those written in mainframe Assembler and COBOL) to run at much lower cost on servers equipped with standard Intel or AMD microprocessors. Such solutions, known as mainframe emulators, are available from T3 Technologies (FLEX) and TurboHercules (Hercules). Because emulators enable mainframe-only operating systems such as IBM's z/OS to run on industry-standard servers simultaneously with modern operating systems such as Linux and Windows Server, they create opportunities for cross-pollination and open up a broad range of innovative new use cases. For example, emulation would make it possible to use a modest array of suitably configured Intel-or-AMD blade servers to provide an off-site or even mobile Disaster Recovery facility for *both* SSA's mainframe applications *and* the Agency's many Windows-based server applications[19]. At a time when SSA is struggling to shore up its inadequate mainframe Disaster Recovery capabilities and is at serious risk of failing to meet its self-imposed deadlines for doing so, mainframe emulation technology could provide dramatic short-term relief. The testimony[20] last April before the House Ways and Means Committee by then-chair of the Social Security Advisory Board Sylvester Schieber regarding the status of SSA's mainframe Disaster Recovery capabilities is particularly relevant in this context:

> "Where does all this leave the agency in terms of operational capacity and its ability to backup data and recover operations as the transition between data centers takes place? Sometime within the next two years the second data center should have sufficient capacity to process some workloads on an ongoing basis as well as provide additional backup and recovery for other critical workloads. This will certainly improve the situation for a period of time. However, by late 2012 when the NCC is at the end of its projected life-cycle, the second data center will most likely need to serve as the agency's primary computing center with disaster recovery once again reliant on commercial hot sites. To date, we are unaware of any efforts the agency has taken to actively pursue alternative recovery scenarios such as contracting for the use of other governmental or commercial hot sites in the event the NCC becomes non-operational… The National Research Council referred to the data stored by SSA as the 'crown jewels.'

[19] At least one vendor of mainframe emulation software, TurboHercules, already offers a Disaster Recovery solution for IBM mainframes based on servers using Intel chips. See www.turbohercules.com.

[20] See "Statement of Sylvester J. Schieber, Chairman Social Security Advisory Board To the Subcommittee on Social Security of the Committee on Ways and Means U.S. House of Representatives April 28, 2009".

> The current two data center strategy affords some assurance that the data are secure and recoverable… In the interim, over the next seven years until the new NCC and the second data center are fully operational, there is a risk that at some point benefit checks could be significantly delayed or not delivered and important data could be lost. Given the economic role that Social Security plays in the lives of a large segment of the American population, I find this situation deeply disturbing."

Although mainframe emulators cannot yet match the performance of the largest IBM mainframes, they are technically mature and have been available for a number of years. If given an opportunity to compete freely for the business of IBM mainframe customers such as SSA, and even if initially confined to niche uses such as Disaster Recovery or less critical business applications, mainframe emulation technologies could be expected to attract fresh investment capital and would likely see their performance and capabilities evolve rapidly. In the long run, the ability of Intel-or-AMD-based mainframe emulation solutions to run mainframe operating systems alongside modern server operating systems on the same hardware may make it possible for large legacy mainframe application portfolios such as SSA's to migrate in a controlled and incremental way to more cost-effective platforms.

Another competitive alternative to IBM leverages IBM's own discriminatory pricing to the benefit of IBM customers. In recent years IBM has used its control of its mainframe architecture to develop an unusual pricing scheme that allows it to sell identical mainframe processors at dramatically different prices depending upon the type of application the customer wishes to run. Standard mainframe processors that run the most valuable legacy applications – chiefly those that IBM's mainframe z/OS operating system and its transaction processing software such as CICS or IMS – are sold at a price that is dozens of times more expensive than processors of equivalent power from chip suppliers such as Intel or AMD. But when a customer wishes to use mainframe processors to run so-called new workloads – which might include programs written in the Java language, off-the-shelf business applications such as SAP, or applications designed for the open source operating system Linux – then IBM reduces the price of those processors by as much as 95%, and also significantly lowers the corresponding software prices. A mainframe processor that has been dedicated to these new workloads is known as a "specialty engine". Mainframe software innovators such as Neon Enterprise Software now offer new technical approaches that enable a greater number of legacy workloads to run on these engines, thus offering users very significant cost savings on both mainframe hardware and software.

Unfortunately, IBM has in recent years sought to suppress competitors such as T3, TurboHercules and Neon by means of intimidation and litigation. In so doing, IBM has departed from its previous thirty-year practice of cooperation with rival manufacturers of IBM-compatible mainframes. Indeed, according to market research firm IDC, as recently as 1999 the so-called "plug compatible mainframe" (PCM) vendors such as Fujitsu-Amdahl and Hitachi controlled nearly 20% of the world market for IBM-

compatible mainframes. Under pressure from American and European antitrust regulators, IBM cooperated for decades with the PCM manufacturers by allowing their customers to purchase the required IBM mainframe operating systems for use on non-IBM hardware. However, IBM's behavior changed dramatically after these vendors withdrew from the market at the time of IBM's transition to 64 bit hardware in 2000. In the following year the Consent Decree that had long barred IBM from engaging in anti-competitive practices ended. This premature end to government oversight removed a fundamental constraint on IBM's competitive behavior. Most regrettably, the change occurred just as a new generation of mainframe-compatible suppliers such as T3 and Hercules were stepping in to fill the shoes of the departed PCM vendors with innovative software-based implementations of the IBM mainframe architecture.

To the extent that SSA and other Federal Agencies are likely to remain dependent on IBM mainframe architectures for many years to come, it is essential that second source suppliers of this type of hardware be allowed to participate in the market. Federal IT buyers should require of IBM as a condition of doing business that it resume its long-standing policy of cooperation with competing mainframe providers on fair and reasonable terms.

It is difficult to estimate precisely how much money SSA could save if it were allowed to procure IBM-compatible mainframe technologies from multiple competing suppliers in addition to IBM. Nevertheless, there are ample grounds for believing that the savings would be very large, undoubtedly measuring in the billions of dollars over the lifetimes of the modernized versions of the Agency's "citizen critical" applications and the new data center facilities it is building to house them. According to the SSAB, roughly 70% of the $686 million the Agency budgeted last year for IT equipment and services was devoted to "infrastructure maintenance just to keep current systems operating"[21]. An additional $350 million will be required to purchase the computers and storage systems required for the Agency's planned new $500 million data center. Even modest incremental reductions in these expenditures would add up to large savings over a period of years.

Moreover, the high percentage of its current IT budget that SSA must devote to keeping its legacy infrastructure running is a serious barrier to its ability to develop urgently needed new applications in a timely manner. In the words of the SSAB, the "large fixed costs of running an outdated system have not allowed SSA much flexibility to shift resources into 'product development'." A critical case in point concerns the Agency's need to develop a broad range of new online services for citizens and beneficiaries. Without timely deployment of these services, the Agency will fall short in its efforts to reduce a huge and growing backlog of disability claims or to cope in the future with the surging workloads caused by the Boomer generation's retirement. In short, while the financial benefits for SSA of a competitive market for

[21] Social Security Advisory Board, *Bridging The Gap: Improving SSA's Public Service Through Technology*, April 2009.

IBM-compatible mainframe technologies would be very significant, the strategic benefits for its "citizen critical" applications would be of even greater consequence.

How SSA can make its citizen data safe for the future

> "A government should never be dependent on a single vendor's technology to use its own information."
>
> - ODF Alliance

What can be done to insure that SSA's information systems deliver on the obligations of openness and universal accessibility?

We recommend the following steps:

First and foremost, the Office of Management and Budget and the Federal Chief Technology Officer should direct SSA to move as expeditiously as possible toward open standards for citizen data stored in programmatic as well as documentary forms. In particular, the Agency must undertake to carry out the modernization of its mainframe database and associated applications in a way that respects the principles of open data access. Above all, this modernization must remove all impediments caused by the use of mainframe technologies to a greatly expanded use of online services to citizens and to the provision of adequate Disaster Recovery facilities for its "citizen critical" data and applications. Congress must provide SSA with the resources necessary to accomplish this important transformation of its technical infrastructures. The SSA itself and its management must promise to pursue this goal with the greatest energy possible and without delay. The SSA must understand that this commitment is made not only to the President and to Congress but to the American people.

Second, Even if it chooses to retain a database architecture based on the mainframe version of IBM's DB2 relational database, SSA should be required to hold all critical citizen data in properly normalized data tables that can be read and modified by any widely used relational database, including non-mainframe databases such as Oracle 11g, Microsoft SQL Server and non-mainframe IBM DB2, as well as open source databases such as MySQL and PostgreSQL. Furthermore, SSA should be required to separate all critical citizen data cleanly and completely from the programming logic of its applications. Although the continued use of COBOL may be acceptable with proper reengineering, the Agency should plan to replace all application programs using the low-level IBM Assembler language which is inextricably tied to mainframe hardware.

Third, In accordance with the recommendation of the National Research Council, citizens should have the right to consolidate their personal Social Security records with other financial and personal information in the computer systems of third party organizations of their choice such as banks, brokerage firms or retirement funds.

Fourth, Acknowledging that it may not be feasible in the short term for SSA to migrate all of its critical applications and citizen data off the mainframe platform, the Agency should be instructed to hold all of its computer system suppliers to a mandatory second source requirement. In particular, SSA should require IBM to resume its practice of technical and commercial cooperation under reasonable and non-discriminatory terms with suppliers of alternative mainframe systems compatible with the IBM mainframe architecture.

Fifth, SSA should implement the SSAB's recent recommendation that it follow the IRS in confiding the assessment of its IT projects to a vendor-neutral third party. Like the IRS, it should insure transparency of its IT investment decisions by making public all votes of its IT Governance Committee and Executive Advisory Board.