Trinidad and Tobago Linux Users Group
c/o Richard Jobity
#15 Marcus Park
El Dorado
Trinidad and Tobago

25th November 2006

Executive Manager, ICT Policy and Stakeholder Engagement.
Information and Communications Technology Division
Ministry of Public Administration and Information
Lord Harris Court
52 Pembroke Street
Port of Spain
Republic of Trinidad and Tobago

Re : Comments on the role of Open Source Software in Trinidad and Tobago, (2006 – 2008)

The Trinidad and Tobago Linux Users Group (TTLUG) is writing to comment on the paper issued by the Ministry of Public Administration and Information entitled "The role of Open Source Software in Trinidad and Tobago (2006 – 2008)", available for comment on the fastforward website.

The group solicited comments on the proposed policy from TTLUG members in three ways:

1. from the TTLUG mailing list (http://tech.groups.yahoo.com/group/ttlug);
2. through a public wiki set up for this purpose (http://ttlug.pbwiki.com); and
3. a TTLUG meeting held on November 4, 2006.

In addition, individual TTLUG members also participated in the TTCS discussions of the policy, including attendance and participation in their meetings.

Comments from all sources were collated to formulate the TTLUG response.

Richard Jobity
President
Trinidad and Tobago Linux Users Group

Consultation Response Form
The role of Open Source Software in Trinidad and Tobago (2006 – 2008)
A Consultation paper

Name	Richard Jobity
Organisation	Trinidad and Tobago Linux Users Group
Postal Address	#15 Marcus Park El Dorado Trinidad and Tobago
Telephone/Fax	(868) 620-5550 / (868) 361-8234
Email Address	richjob@jobity.com
Website	http://www.ttlug.org
Title of Consultative Document	**The role of Open Source Software in Trinidad and Tobago (2006 – 2008)** **A Consultation paper**

Some definitions used in this document.

Free /Libre Software (also Free and Open Source Software (FOSS), also Free, Libre and Open Source Software (FLOSS)):[1] is software which can be used, copied, studied, modified and redistributed with little or no restriction. Freedom from such restrictions is central to the concept, with the opposite of free software being proprietary software (a distinction unrelated to whether a fee is charged). The usual way for software to be distributed as free software is for the software to be licensed to the recipient with a free software license (or be in the public domain), and the source code of the software to be made available (for a compiled language).

FUD: Fear, uncertainty, and doubt (FUD) is a sales or marketing strategy of disseminating negative (and vague) information on a competitor's product. The term originated to describe misinformation tactics in the computer hardware industry and has since been used more broadly. FUD is a manifestation of the appeal to fear.

GORTT: Government of the Republic of Trinidad and Tobago

GPL:[2] The GNU General Public License (GNU GPL or simply GPL) is a widely used free software license, originally written by Richard Stallman for the GNU project.The GPL grants the recipients of a computer program the rights of the free software definition and uses copyleft to insure the freedoms are preserved, even when the work is changed or added to.

1 http://en.wikipedia.org/wiki/Free_software
2 http://en.wikipedia.org/wiki/GNU_General_Public_License

Open Document Format (ODF):[3] Short for the OASIS Open Document Format for Office Applications, ODF is an open format for saving and exchanging office documents such as memos, reports, books, spreadsheets, databases, charts, and presentations. This standard was developed by the OASIS industry consortium and based upon the XML format originally created by OpenOffice.org. ODF was approved as an OASIS standard on May 1, 2005, and was approved for release as an ISO and IEC International Standard (ISO/IEC 26300) on May 8, 2006. The OpenDocument standard has been developed by a variety of organizations and is publicly accessible. This means it can be implemented into any system, be it free software/open source or a closed proprietary product, without royalties. The OpenDocument format is intended to provide an open alternative to proprietary document formats so organizations and individuals can avoid being locked in to a single vendor.

ODF is the first standard for editable office documents that has been vetted by an independent recognized standardization body.

Open Source Software (OSS):[4] Open-source software is computer software whose source code is available under a copyright license that permits users to study, change, and improve the software, and to redistribute it in modified or unmodified form. It is the most prominent example of open source development.

Proprietary software:[5] -Proprietary software has rules defined by its creators or owners. This kind of license does not come under purview of any country level or international law. The terms and conditions provided are defined only by the owner and/or creator.

Public domain software: Public domain comprises the body of knowledge and innovation (especially creative works such as writing, art, music, and inventions) in relation to which no person or other legal entity can establish or maintain proprietary interests within a particular legal jurisdiction. This body of information and creativity is considered to be part of a common cultural and intellectual heritage, which, in general, anyone may use or exploit, whether for commercial or non-commercial purposes. As such,. public domain software is software where the author has abandoned the copyright. Since public-domain software lacks copyright protection, it may be freely incorporated into any work, whether proprietary or free.

TTLUG: Trinidad and Tobago Linux Users Group

3 http://en.wikipedia.org/wiki/OpenDocument
4 http://en.wikipedia.org/wiki/Open_source_software
5 http://simple.wikipedia.org/wiki/Proprietary_license

General comments

The TTLUG welcomes this initiative by the GORTT to solicit input on Open Source software in Trinidad and Tobago and is happy to provide comments. However, the document put out for comment as presented is a hodgepodge of incorrect facts, incomplete, improperly cited and plagiarized references, and opinions and inferences with little empirical evidence to back the conclusions. In addition, the discussion of the use of OSS does raise some other critical IT issues that deserve greater focus which have either not been raised at all in the paper or mentioned only in passing. As one example, **it is inconceivable that a government policy dealing with free and open source could be written without a proper discussion of free and open source licenses, including the GNU GPL, the Apache Software License and the BSD license. This alone should cause any conclusions drawn in the document to be taken with a grain of salt.** More importantly, the initiative into open source by the Ministry seems to conflict with existing ICT policies currently being implemented by other ministries, and also seems to be in diametric opposition to existing Government information technology policy. The TTLUG has a very real concern that after all the analysis, any initiative proposed will be overridden and/or negated by senior public servants or politicians with a vested "interest" in maintaining the status quo, and that the public view will not be considered.

6. Background

Paragraph 1. (page 4)

The first paragraph attempts to place the government's use of ICT in historical context. This is commendable except for the fact that most of the cited evidence is inaccurate and is an attempted whitewashing of the historical record. Use of software in the government was on an ad-hoc basis in the early 1990s with

several different software packages being used, dependent on the needs of the individual department. Moreover, like the rest of the country the "ICT thrust" was based on the use of illegally copied proprietary software, the use of which was rampant through the government.

> *To rapidly facilitate this thrust, the marketplace rapidly migrated towards widespread use of licenced software packages in the office place.*

This is incorrect. Existing copyright laws mean that ALL software including software under free software licenses and open source licenses are licensed unless the creator of said software explicitly places the software in the public domain. Existing copyright law states that the creator has the right to license the software in any manner of his choosing, and this can include any proprietary license or any open source/free software license of their choosing. In addition, all software can be commercially sold, and moreover, proprietary software also comes in a variety of licenses.

> *This deference to popular, off the shelf licensed software persists as something of a "default" up to today.*

This scenario, while **technically** accurate, is more because of the enterprise agreements the that Government signed with a monopoly software provider. Anecdotal evidence suggests that this occurred after an audit by said software provider found the government using their software illegally. The alternative to massive fines (at the time) was the government licensing the company's sofware in a perpetual licensing agreement rolled over, seemingly without any transparent tendering process and with specification criteria slanted to one operating system. Globally, the use of proprietary software only became a default after the mid 1970s[6]. Before that, source code was provided with computers.

6 The letter referred to in http://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists is generally recognized as the beginning of the use of proprietary licenses, where products was distributed without source code.

Paragraph 4 (page 4)

Paragraph 4 attempts to place fastforwards's philosophy in the context of global standards, but only manages to muddy the distinction between open *standards*[7] *(publicly open and maintainable standards, allowing for maximum hardware and software compatibility)*, open source *software* and open source *methodology*[8] (practices in production and development that promote access to the end product's source materials). The amended document should seek to make clear the distinction in order to avoid confusion among the readers.

Paragraph 5, (page4-5)

The document states,

> *The argued advantage of licenced or closed source products is inimically tied to the perceptions such as:*
>
> - *the ease of training in these licensed packages;*
> - *the ability to seamlessly integrate with the commercial packages which public servants and consultants may already be using otherwise; and*
> - *suggested cost and maintenance advantages.*

This argument exists in perception rather than fact. Regarding the first point, there is no perceptible difference in training someone in a software concept embodied in an open source package and the same concept as expressed in a proprietary package. Ease of software training is about the same for someone not previously trained in any package. The second point implies that there are integration problems vis a vis free and open source packages and proprietary counterparts, but in actuality, this may actually be the fault of the makers of proprietary software who refuse to implement free and fully documented file standards, attempt to make proprietary open standards[9] and use their market

7 http://en.wikipedia.org/wiki/Open_standards
8 http://en.wikipedia.org/wiki/Open_source
9 http://www.usdoj.gov/atr/cases/ms_tuncom/major/mtc-00029523.htm (search for Kerberos)

position to throttle competitors[10]. In response to interoperability needs of the users of FOSS packages, the software is coded to cater to the requirements of proprietary packages. For example, the mail, scheduling and calendaring application Evolution supports connection to a Microsoft Exchange server. Similarly, OpenOffice.org, apart from offering the ability to save files in the ISO-approved ODF format, also has the ability to save files in all of Microsoft's proprietary document formats from Office 1995 Office 2003, usually with a higher level of fidelity for older Microsoft formats than the most current offerings from the Microsoft office suite. More recently, hardware and software manufacturers have been offering binary or source code interoprerability with FOSS projects – for example, nVidia offers modern graphic card drivers that sometimes have more features than their Windows equivalents. Oracle runs on Linux as its preferred platform, and even Microsoft customers have demanded interoperability to the point where Microsoft has entered into partnerships with Linux vendors to offer interoperabilty agreements.

Paragraph 6 (page 5)

The phrasing and wording of this paragraph can easily be combined into paragraph 7, immediately below.

Paragraph 7 (page 5)

The author again gives a needlessly muddled definition of open source software, and further compounds the inaccuracy by stating that

> *OSS typically provides others with the ability to develop a new version of the software, port it to other operating systems and processor architectures, share it with others, or even productize and sell it.*

There is nothing in the definition of OSS or even free software that prohibits

10http://tinyurl.com/4ds4f

people from charging what the market will bear for an open source product. Moreover, that fact that a piece of software is open source in and of itself does not confer the ability of the software to be ported to different architectures, or prohibit development. The Opera browser was sold for many years, in an environment where all other browsers were free of cost, and in terms of portability, opera is available on multiple computer architectures (Windows and Linux) and even on cellular phones. Generally speaking, this paragraph can be simplified greatly, while maintaining the core message.

Section 7. Open Source Software – Threat or Promise (page 6)

The first paragraph states that

> *Trinidad and Tobago plans to become a knowledge-based society by 2008, therefore software will rapidly become one of the most fundamental building blocks of human interaction and activity.*

This appears to be a straight "cutting and pasting" from the ICT4D National Policy (2003) fastforward document, the summary of which states,

> *Formulated in 2003 and focusing on human resources, economy and finance, government, infrastructure, and the legal sector, the policy lays out an action plan for achievement of its objectives by 2008*.[11]

Given the GORTT's preoccupation with other projects and the shortage of all categories of labour consistent with a rapidly expanding economy, this deadline seems quite ambitious. In addition, international experience suggests that the remaining time period for implementation of OSS from conception through pilot projects to roll out (with the original schedule, December 2006 to December 2008, approximately 25 months) is far too short. The only thing that could be

11 http://www.comminit.com/trends/ictpolicies/ictpolicies-17.html

reliably implemented given that time is failure. The TTLUG suggests caution. Any defined relationship over the next 1.2 years makes no sense, as there will be an adoption period. We seem to be expecting the role of FLOS to change in 1.2 years, yet the new passports promised to citizens years ago are still not available, to cite an example. Any relationship beyond the 2006-2008 time frame is good. Promoting the use of FLOS 'when proprietary solutions have no competitive advantage' is approaching the issue from the wrong angle. We need to step back and look at all solutions 'open' and 'closed' and give the edge to which one gives the best ROI, widest accessibility, stability, and gets the job done properly.

> *"therefore software will rapidly become one of the most fundamental building blocks of human interaction and activity."*

Unlikely. While software can ease human interaction and activity, it complements the existing building blocks, rather than usurping the role of other fundamental tools.

Paragraph 2 (page 6)

It is clear that this part of the paper is authored by someone else, as this definition of open source given in this paragraph is more in line with international norms.

> *Trinidad and Tobago now needs to consider if heading in this direction is feasible and appropriate given our specific context. Some claim that OSS confers a promise of better software and independence from perceived monopolistic behaviour and vendor capture. Such a promise would also have far reaching consequences for employment and opportunity.*

The TTLUG believes that the question is not if heading in this direction is feasible, but how much. While we believe that OSS does confer higher quality and relative freedom from vendor lock-in and monopoly abuse, the GORTT study would be greatly enhanced by a relevant citation.

7.1 The Common Threats of Open Source Software

Section 7.1 seems to be an uncredited paraphrase of an article[12] which coincidentally covers the same points. With regard to the security thread cited by large proprietary companies (covered, for example, by Microsoft's letter to Congressman Edgar Villanueva Nuñez of Peru[13]) security problems occur in all code- both proprietary and open source. The fact that the source code is open does not make the code inherently better or more or less secure – rather, the quality of the code and the quality of the code maintainers determine how well the code operates in a real-world environment.[14] Peruvian Congressman Edgar Villanueva makes the definitive rebuttal to Microsoft FUD in his reply to MS Peru.[15]

7.1.1 The security threat

Given the succinct summary of 7.1.1 and 7.1.2 in 7.1, one questions the need to further expand on them. Moreover, section 7.1.1 seems to be concentrating on opinions, and not simply stating facts. Moreover, the second paragraph of 7.1.1 introduces the possibility of a ban on OSS in the information Security sector – a strange insertion considering that this paper is a discussion paper, without policy discussions taking place at this point.

In the software world, it is generally known that the best programmers in the world work for NASA. It is interesting therefore to see that NASA uses open source software. While NASA could afford to purchase any software package available, NASA not only purchases and uses open source, NASA produces open

12 http://freebies.about.com/library/weekly/aa061602a.htm
13 http://www.opensource.org/docs/msFUD_to_peru.php
14 Noted security expert Bruce Scheier claims that "The best algorithms we have are ones that have been made public, have been attacked by the world's best cryptographers for years, and are still unbreakable." Applied Cryptography (Second Edition); page 7
15 http://www.opensource.org/docs/peru_and_ms.php

source.[16] In the US Department of Defense (home to the most security-conscious hackers anywhere), open source is a valued component of the DoD arsenal.

According to the DOD Open Technology roadmap[17],

> *The national security implications of open technology development (OTD) are clear: increased technological agility for warfighters, more robust and competitive options for program managers, and higher levels of accountability in the defense industrial base. DoD needs to use open technology design and development methodologies to increase the speed at which military systems are delivered to the warfighter, and accelerate the development of new, adaptive capabilities that leverage DoD's massive investments in software infrastructure.*

OSS and open source development methodologies are important to the national security and national interest of the U.S. for the following reasons:

- Enhances agility of IT industries to more rapidly adapt and change to user needed capabilities.
- Strengthens the industrial base by not protecting industry from competition. Makes industry more likely to compete on ideas and execution versus product lock-in.
- Adoption recognizes a change in our position with regard to balance of trade of IT

Finally, the original DoD debate on the use of OSS[18] dealt with security issues in great detail, including the Mitre report on the use of OSS in the US Department of Defense[19],

16 http://opensource.arc.nasa.gov/
17 http://www.acq.osd.mil/actd/articles/OTDRoadmapFinal.pdf
18 Summarized at http://www.terrybollinger.com/index.html#open_source_reports
19 This debate is summarized on Wikipedia at http://en.wikipedia.org/wiki/Use_of_Free_and_Open_Source_Software_(FOSS)_in_the_U.S._Department_of_Defense

> *The main conclusion of the analysis was that FOSS software plays a more critical role in the DoD than has generally been recognized. FOSS applications are most important in four broad areas: Infrastructure Support, Software Development, Security, and Research.* ***One unexpected result was the degree to which Security depends on FOSS. Banning FOSS would remove certain types of infrastructure components (e.g., OpenBSD) that currently help support network security. It would also limit DoD access to—and overall expertise in—the use of powerful FOSS analysis and detection applications that hostile groups could use to help stage cyberattacks.*** *(emphasis mine) Finally, it would remove the demonstrated ability of FOSS applications to be updated rapidly in response to new types of cyberattack. Taken together, these factors imply that banning FOSS would have immediate, broad, and strongly negative impacts on the ability of many sensitive and security-focused DoD groups to defend against cyberattacks.*

Trinidad and Tobago could do well to emulate this example.

7.1.2 The economic threat

While the heading was changed to "threats" from "impacts", this paragraph suffers from the same problem as the preceding paragraph.

> *in particular, the free, libre open source software (FLOSS) model are anti-competitive and may cause a loss of jobs and corporate revenues. They also point out that open software is not truly free and will actually cost the government/ corporate body more in training and support.*

Once again, opinions are stated and used to draw conclusions without supporting evidence. Again, the Villanueva letter prove important evidence to rebut and counteract FUD. More critically, the framers of the document concentrate on economic impact, but have not discussed ownership and software licensing. It appears that the MPAI document framers have not understood clearly that the point of FLOSS is freedom from licensing restrictions, and this has a very

significant positive impact, since local people can build new and customized applications built on world-class code, and fix (or have someone fix) their own code, and not have to depend on the dubious mercies of a foreign company that may not have national interests at heart. They are, in fact, diverting the discussion away from the licensing issue, only mentioning it in passing.

With regard to training and support, it is true that the existing training in Trinidad and Tobago concentrates on proprietary software, but this is solely as a result of market forces and work requirements promoting particular types of IT skills. To suggest that open source has unique problems regarding training of personnel is a red herring. As the need for local training on proprietary software was answered by the rise of local training centres, once FOSS becomes more popular, institutions will respond to the demand by introducing FOSS courses into their curricula. Bordercomm and SBCS already provide FOSS courses and the University of the West Indies and Petrotrin have run FOSS courses in the recent past. In addition to local training leading to international certifications, several organizations and companies offer courses and certification in FOSS. These include Red Hat, IBM, Novell/Suse, HP, Canonical, the Linux Professional Institute and CompTIA.

Finally, the two main things that separate open source from proprietary applications are the license, and the development model. Since open source licenses allows users full access to the source code and the right to re-distribution, the question really becomes "Why would anybody want to use closed-source and proprietary applications in the long run?"

The only "real" reason becomes support issues. Businessmen, and governments, in particular, like to have someone else to blame. Open source software projects have addressed this need with a variety of options, including paid support, such as service contracts (Red Hat, Canonical (Ubuntu)), free support such as mailing lists and forums, user groups, mixed-mode licensing options (offering products

both under open source licenses and commercial licenses, as in the cases of MySQL, Qt, and RedHat) It should be noted however, that these commercial licenses from open source entities do not normally restrict re-distribution and modification.

7.1.3 The Development Opportunity

Licensing and ownership issues are key to fostering a climate where software development and production can take place. If there is ambivalence on this issue, there is no possibility of a local software development industry taking root. Neither open nor closed source can find encouragement in the absence of legislative protection of licenses. We will always have to import software in the presence of such ambiguity. In fact, a move to real indigenous software development is desirable, as the absence of such simply ensures that we remain a nation of software consumers or, more accurately, software copyright infringers. The use of open source software and its insistence on the right to distribution almost by definition sidesteps the problem of software piracy by tapping into the basic human impulse to share.

> *Open source software is also inherently more suited to educational environments because its inner logic - the source code - can be directly manipulated by students. With its inner parts visible, users can choose to learn how the software works and then share and develop that knowledge. Proprietary software, by contrast, is inherently "unknowable" because its inner architecture is a trade secret.*

This developmental opportunity would be far more realizable if our education systems (secondary & tertiary) used open source software in their teachings. As currently established, UWI does not use much open source software in it's teachings. Currently, undergraduate students in the studying computing are exposed to open source code only in the Operating Systems course, though to be fair, the Department of Mathematics and Computer Science are more open source friendly than they were five years ago. At secondary level, proprietary software

is taught as an integral part of the CXC Information technology syllabus, so students end up learning proprietary software packages, not concepts easily transferable to other types of software.

The huge advantage that open source code has over its closed source counterpart is that the open source community sees itself as a meritocracy. Contributions that are of little value are discarded and open source projects that do not attract a community end up as 'orphan code'. This approach may seem disconcerting at first glance, but it is, in fact, the willingness to abandon a project and start in a new direction (forking), that leads to the high value of open source versus closed source applications. Open source project leaders care passionately about the value of their code, and reject modifications which they see as reducing the quality of the finished product. Projects like FreeBSD and the Apache projects are relentless in their hunts for bugs. Consequently, the stability of these systems is among the highest for software on the planet.

The freedom from commercial considerations allows project leaders to take the time necessary to eliminate bugs from their projects, through bug tracking and regular security and software audits. It is practically an open source standard that projects maintain a database of bugs that any user can add to. This transparency allows the user base to participate in the improvement of the software. Bugzilla's installation list[20] reads like a who's who of the open source movement.

7.2 OSS and Market Share

This section appears to be a direct paraphrasing of a section of David A. Wheeler's essay on FLOSS, "Why OSS/FS? Look at the Numbers!"[21] and as such, there is no real issue with the data presented therein. It is to be hoped that the framers of the MPAI OSS document cite more recent statistics in their analysis in

20 http://www.bugzilla.org/installation-list/
21 http://www.dwheeler.com/oss_fs_why.html#market_share

any future redraft of the policy.

7.3 Reliability

This section also appears to be a direct paraphrasing of a section of David A. Wheeler's essay on FLOSS, "Why OSS/FS? Look at the Numbers!"[22] However, we note that the MPAI study quotes extensively from "Fuzz Revisited"[23], a paper done in 1995 for Unix-based applications. Between then and now, open source development methodology has ensured that current systems are far more reliable. It would be desirable to see some updated metrics, but the most recent analysis on the site is for Mac OS X, done in 2006[24]. That study concludes:

> An optimistic view of software evolution would be that, as we learn more about the software development and engineering process, code should naturally get better. The pessimist (or perhaps the realist) would note that the commonly used programming languages and operating systems are not notably different from those that we used twenty years ago. In addition, software packages are providing more features and therefore are getting more complex. In such a view of the world, it is not surprising that the reliability of GUI based applications is not improving, but instead seems to be getting worse.

7.4 Governments and OSS

This section also appears to be a direct paraphrasing of a section of David A. Wheeler's essay on FLOSS, "Why OSS/FS? Look at the Numbers!"[25]. While the TTLUG generally has no issue with the arguments advanced in this section and its subheadings, we note with some dismay that certain issues were not highlighted. The document is silent on the role that OSS can play in safeguarding constitutional rights of the citizenry, privacy issues, research and technology, job

22 http://www.dwheeler.com/oss_fs_why.html#reliability
23 http://www.cs.wisc.edu/~bart/fuzz/fuzz.html
24 ftp://ftp.cs.wisc.edu/paradyn/technical_papers/Fuzz-MacOS.pdf
25 Copied from http://www.dwheeler.com/oss_fs_why.html#governments

creation, national security and this would seem to preclude the overuse of proprietary software in certain endeavours.[26] According to Tony Stanco, as provided to the New York City Council's Select Committee on Technology in Government,[27] (emphasis in quotes are mine)

> Governments are special entities and their functions and operations can be at odds with proprietary software applications that are developed for a multiple of purposes. **Governments have special obligations to protect the integrity, confidentiality and accessibility of public information throughout time like no other entity in society.** Therefore, **storing and retrieving government data through secret and proprietary data formats tied to a single provider is especially problematic, since the usability, maintenance and permanence of government data should not depend on the goodwill or financial viability of commercial suppliers.**
>
> Furthermore, **citizens have a right to transparency in public acts, which may be hampered by secret, proprietary software.** A clear example of this is e-voting software. I expect no one would seriously defend the right of proprietary software companies to prevent political candidates from inspecting the software that tallies the votes in elections. There are many other public acts that fall into the same category. So many in fact that the onus should rightly be placed on companies to justify the use of proprietary software in purely governmental settings.
>
> Privacy
>
> There is a constitutional right to privacy, and **it is incumbent on government to set rules to protect the privacy of its citizens. Software that may transmit private data to, or allow control and modification of computer systems by, third parties without the explicit consent of the user is a violation of the citizen's right to privacy.** It is disingenuous to argue, as Open Source opponents often do, that the market will sufficiently protect the rights of citizens in these circumstances. **Software follows the principle of "network effects" where, after a certain tipping point, all consumers lose their freedom of choice and**

26 http://www.egovos.org/Resources/Testimony. Also covered thoroughly in the Villanueva letter to Microsoft Peru, cited earlier.

27 Ibid.

> **are herded into using the same product for the sake of interoperability. The existence of monopoly situations in software also work to restrict freedom of choice, further limiting the protective effects of a purely market-based solution. As a result, government intervention is appropriate to protect the privacy rights of its citizens.**

If the government is interested in any research and technology, they would do well to consider Open Source. Stanco states,[28]

> The Open Source method is analogous to the scientific method, where researchers share information and results, and are not hampered by having to constantly "reinvent the wheel." Additionally, Open Source researchers do not have to deal with expensive and restrictive licensing terms, which arbitrarily preclude the involvement of talented people. This creates a very low threshold to get into the research and development of projects, allowing smaller schools and even industrious individuals to participate.

Given the Government's investments in egovernment and IT infrastructure through fastforward, it is hoped that open source will be given more than token consideration.

The same thing goes for education.[29]

> Open Source is a superior way to educate the next generation of IT professionals. With Open Source, the developers see and study the actual code running real world systems, rather than working with stripped-down "toys" designed merely for educational purposes. Many developers have recounted that they learn best by trying and watching what happens in the program as it runs. This should not be surprising at all, since this was how developers learned the craft before the 1980's when the closed software industry arose. Open Source is just returning software to its free and open roots.
>
> It should also be noted that Open Source has marvelous outreach programs run by

28 Ibid.
29 Ibid.

> community groups in most cities around the world. It is very common for teenagers, IT students and experienced professionals to attend free Open Source events to share ideas, software and programming skills.

Open Source can also have a role in job creation. The business model for Open Source software is based on highly technical and specialized services, similar to law, medicine or engineering. Moving government software systems to Open Source where practical means that there will be more local, high-paying IT jobs for integrators and consultants. Furthermore, spin-off economic multiplier effect can benefit others by keeping software dollars in the local economy, increasing the country's potential tax base.

Finally, any government concerned with national security issues would do well to consider open source systems. Stanco states,[30]

> The open secret in the defense and intelligence communities around the world is that Open Source is the preferred software for secure systems. These groups don't trust software that they can't study and compile themselves, because of concerns over bugs and "spyware", and therefore would rather use Open Source software for their sensitive and classified systems.

The analysis would also be greatly facilitated by an evaluation of international experience. An (incomplete) list of government and public initiatives into FOSS can be sourced here.[31]

Paragraph 5, page 12

> "The United States federal government has a policy of neutrality; they choose proprietary or OSS programs simply considering costs and other traditional measures."

30 Ibid.
31 http://europa.eu.int/information_society/activities/opensource/cases/index_en.htm

Government's focus should be on ensuring proper public policy is in place. These policies should include that public documents adhere to open standards and not be tied to proprietary formats, and are secured against theft, loss or unauthorized access. Whether the software is open-source or closed-source, it should meet these fundamental requirements. Open formats ensure data longevity (the ability to access data years in the future) and avoid vendor lock-in. Open Document Format (ODF) is a great format (ISO-approved), and would be even better if more applications started supporting it. However, if a recommendation to move towards ODF is made, who follows through with the recommendation? Practically, we need to use what works while ensuring that as much as the nation's data reside in open formats.

Neutrality is potentially advantageous because all options are considered before making a decision: cost of software, cost of implementation, time taken to deploy, cost of hardware, cost of training, security, ease of learning etc. On the other hand, neutrality can be potentially disadvantageous because it relies on all options being considered. Because most IT personnel are groomed in Microsoft-centric products, they gravitate towards products that they are familiar with rather than examining all solutions. Many will avoid leaving their comfort zone because they fear change and having to learn new technologies. This may be the real sticking point here.

7.5 Developing Countries and Open Source Software

In Paragraph 1 the TTLUG notes with some amusement that while **LinuxAsia** took place in New Delhi in 2005, the only mention of **LinxAsia** anywhere is in the MPAI policy. Be that as it may, the TTLUG has no significant issue with the writeup concerning developing countries and OSS, apart from the lack of citations. However, the paragraph on fastforward and its agenda may be better slotted in the following section, on Free, Libre Open Source Software and NICT

Strategies.

However, the TTLUG is also mindful of the retraining thrust (and one-time retraining costs to be incurred) that must take place if the fastforward agenda is to include use of FLOSS. However, open source software does offer significant cost savings when compared to proprietary software. This should be considered especially if computers are to be deployed to communities and schools throughout Trinidad and Tobago. Given the implied fastforward objectives of capacity development and wider developmental opportunities, the use of Open Source software by the Government of Trinidad and Tobago would be the best choice. By adopting Open Source software the GOTT can legally pursue this objective without fear of licensing costs.

Open Source promotes open formats, is multi-platform (meaning such software can run on proprietary operating systems (such as Microsoft Windows and MacOS X) as well as open source operating systems (such as Linux and FreeBSD)) and this means that such multi-platform open source applications can be deployed in existing IT infrastructure without requiring significant hardware and software changes.

7.6 Free, Libre Open Source Software and NICT Strategies

While no one acquainted with FLOSS will doubt its potential, the question becomes: who will execute the NICT strategy and implement open source strategies in Trinidad and Tobago? Local OSS experts are few, mostly self-taught and tend to be in very high demand. An improper execution or botched OSS program could stymie the widespread adoption of open source. As a result, any local ICT initiative should be well budgeted, with sufficient time given to ensure success and to deal with the inertia issue. Moreover, this is potentially significant enough that widespread buy-in should be encouraged, so the programme becomes somewhat insulated from the political cycle. Ultimately, the TTLUG's

concern is over whether we become a nation of software consumers or software producers. If there is ambivalence on licensing, patent and ownership issue, there is no possibility of a sustainable local software development industry taking root. Neither open nor closed source can find encouragement in the absence of legislative protection of licenses. We will always have to import software in the presence of such ambiguity. Apart from the GORTT's international signing of copyright conventions, e.g., the Berne convention, it would clarify matters if fastforward made explicit mention of the GORTT's real stance on the license issue. As it stands currently, it appears that the Government selectively enforces copyrights in certain cases (local music artistes) and ignores it in others (widespread illegal copying of movies and foreign music) and in fact may indirectly benefit from illegally copied content through tax revenues and VAT paid by "pirate" video clubs.

7.7 Policy Options for Free, Libre and Open Source Software

To its credit, the MPAI outlines the options available. The TTLUG believes that while a balanced playing field and a policy of software neutrality is best, the simple truth is that the existing conditions of software procurement and existing ICT policy is heavily tilted towards proprietary software. As a result, to achieve long-term balance, OSS-leaning policies (or policies seen as such) should be promoted. It is the TTLUG's belief that such intervention should not be heavy-handed and direct, but should more be designed to create an IT ecosystem as favourable to open source solutions as it is to proprietary solutions. Open source software should be mandated only under certain conditions, such as national security concerns. The GORTT should decide if they wish to follow its own model, or adopt elements of FLOSS policy from other nations. This also is important enough such that consensus from all quarters should be sought so the initiatives have protection from the vagaries of the political cycle.

8 Key points of consultation

Key point 1

From your understanding, do you agree with the above analysis?

The model offered by OSS conforms to a different social and economic model on licenses, patents, and copyrights and this model should be fully analyzed. While the discussion offered in the Consultation document does cover some of the relevant issues in an accurate and adequate way, the discussion of the use of open source and open standards raises some critical issues which have either not been raised in this paper or have been mentioned only in passing. We believe that these areas deserve greater focus.

These areas include

- Industry development. In Trinidad and Tobago currently, there is a small software development industry, limited to niche markets like payroll software, customs brokerage software, local websites etc. There is also rampant copying of commercial software and little if no enforcement of software licenses. The growth of local software development is hampered significantly by the lack of protection which should be provided by the enforcement of copyright and licensing on software. Like all other software, FOSS is based on copyright and the rights conferred by the GPL and similar licenses.
- Training. This has been dealt with previously in the submission.
- ICT 4 Developement (http://www.ict4d.org.uk/) - these include issues of access; localisation of software to local needs, the ability to customise the software, accessibility issues and access for the differently abled, etc
- integration with other software (use of open source on proprietary operating systems). There are problems with integration. But this is true of all software. Open source approaches these issues by promoting open formats to improve integration. It is, in fact, proprietary software vendors who refuse to support such initiatives. The reason that integration problems still exist is the explicit refusal of large proprietary software vendors to cooperate with existing

standards or to release the information necessary for FOSS producers to co-operate with their standards. The problem exists because commercial software vendors create it, in order to "protect their turf". Even so, FOSS producers have responded by independently reverse-engineering closed formats and protocols. For example, OpenOffice.org both reads and writes Microsoft formats. The Samba networking suite provides connectivity between Windows networks and non-Windows networks.

It is not to be supposed that integration problems are insurmountable. The local Trinidad & Tobago Computer Society provides at a nominal cost a CD full of quality FOSS software which works perfectly under Microsoft Windows, including Apache, OpenOffice.org, MySQL, PHP, the Firefox browser, Mozilla Thunderbird and the Mozilla Seamonkey internet suite.

General ease of use in FOSS applications is a problem which have been identified and is being aggressively tackled. It is an understatement to say that the harshest critic of FOSS is the FOSS community itself. It is *precisely* this constant self-criticism which has improved FOSS to the point where it is having such a world-wide impact.

- Patents, copyrights and national security issues. Also analyzed in this response.

Key point 2

*Do you think **fastforward**, the NICT Plan should encourage the use of open source software, and accept the associated 'risks' when very sensitive information is at stake?*

Yes, the NICT Plan should encourage the use of OSS as the national benefits are clear. Where there are risks identified in the document, the GORTT through fastforward can identify what these risks are and suggest the mechanisms by which these risks can be mitigated or alleviated. The NICT Plan should also include some ideas for when the Govt is also the producer of software - the Govt should also decide to commit to the OSS model to gain the benefits from using a FLOSS strategy. However, the phrasing seems to imply that there are risks

unique to FOSS software that are not present in proprietary software. **This is not so**. In fact, public domain or proprietary software is as likely or more likely than open source software to be subject to such risks, including patent violations. In cases where patent or copyright risks are a concern, this can be alleviated by using FLOSS software from an enterprise producer, such as Red Hat or IBM.

Key point 3

From your understanding, which of the above perspectives is accurate?

Which above perspectives? Key point 2 is accepted conditionally as is Key point 1.

Key point 4.

What do you think should be the primary goals guiding decisions for or against the use of open source software in Trinidad and Tobago?

Given that the use of Open Source software confers definite advantages without associated risks, there are several good reasons guiding the decision for the use and promotion of open source software in Trinidad and Tobago, and these have been expounded on in the comments. The primary goals should include software independence, security and transparency. While costs for OSS are indeed lower, and the use of OSS gives you a far greater total cost of investment, lower cost is not and should not be the motivating factor. While licensing costs might be less overall, there will be one-time costs associated with training and installation, which could be paid to local professionals, keeping the money in the local economy and improving the local services balance.

Other factors that might be considered include the encouragement of a local software and software services industry, encouraging the expansion of local ICT, freedom from vendor lock-in and the possibility of "orphan" proprietary software, freedom of critical national and citizen data from being held hostage by

proprietary formats. The creation of employment in the local ICT industry for people with OSS skills (trainers, programmers, software developers, project and community managers, system administrators, computer technicians and other skills) is also a useful benefit, as these skills are transferable both between countries (the Caribbean region is expected to use far more OSS in the future as other countries move from proprietary systems to open source systems) and across operating systems.

Key point 5

Given the above data, and your understanding of the current state of the ICT sector in Trinidad and Tobago, is there a potential to develop numerous niches and cost saving strategies through the implementation of OSS?

As currently obtains, the greatest untold story in the local ICT industry is the use of OSS. While OSS on the desktop is still limited, more and more individuals and organizations are using OSS tools and systems on the back end to run their systems more efficiently and get greater value for their ICT dollar. A lead or facilitation role by the GORTT will also serve to further spur OSS adoption. However, the issue of execution of strategy, as well as personnel to carry out the strategy, pose a significant potential impediment to OSS adoption. At best, however, it will only postpone the inevitable widespread adoption of OSS. The government can either lead the move, follow, or facilitate from the sidelines. Yes, though it must be recognized that at the end of the day, FOSS is a tool to make you more efficient. It is not a silver bullet or a cure-all for your IT skills. As a result, as with any other software, proper planning is needed to make the software reach its fullest potential.

Key Point 6

In your opinion, what new structures are needed to encourage niches of activity through OSS?

There should be some structure which would encourage the growth of local software development skills. This can either be at the primary/secondary level (basic programming concepts) or at the tertiary and vocational institution level (practical courses, system administration and the like). As part of the facilitation process, the GORTT may have to embark on a one-time retraining/training programme in the short term.

Key point 7

Considering the comments above, should the open source software model follow the FLOSS philosophy? Please provide context to your response.

This question seems strange, as OSS software by definition DOES follow the FLOSS philosophy. As it work well in other places, given proper analysis and execution of the issues, the answer is yes. In formulating the overall model for the country, allowance should be made for sufficient flexibility to allow for unforeseen institutional and other events specific and relevnt to Trinidad and Tobago.

Key point 8

Considering the points above, and your contributions to key points 5 and 6 above, what role if any do you think the Government of Trinidad and Tobago should play with regard to the open source industry, as opposed to the proprietary software industry, in the country?

Even with widespread adoption of FLOSS, the Government should not think that its use of OSS is going to kill the local or international proprietary software industry. Many service providers deal with a mix of OSS and proprietary tools, and even the most reluctant holdouts on the side of proprietary software have recognized the need for coexisting with OSS, because their customers work with OSS and demand interoperability. Even with widespread OSS use, it is likely that the GORTT is going to need proprietary software in some areas.

The government should seek to ensure that the software industry, both local and proprietary exist on a level playing field. Given that such is not now the case despite what advocates of proprietary software may say, the GORTT does have a responsibility to ensure that the playing field becomes level, through whatever means necessary. While OSS software can fulfill the same stringent procurement criteria as the best of proprietary software, the government needs to ensure that procurement tenders and software requirements are not written to lock out non-proprietary options, such as adherence to open formats, no proprietary lock-in. While the ongoing reform of the public sector procurement process should make the tendering procedure a bit more transparent, IT-related tenders are still sufficiently opaque to be a cause for concern. Long term IT policy should seek to combine optimum operation of IT resources with the encouragement of open options, that free you from dependence on one vendor who may not have the national interest at heart.

Proper policy must be in place first before anything else first can be done.

Key point 9

Should Government look primarily at formal (direct) or informal (non-direct) approaches to the question of software models (proprietary or open) within the ICT sector?

There should be a mix of both approaches, since informal use in and of itself will not encourage a shift to OSS. A clearly articulated policy and efficiently executed strategy that identifies the areas in which OSS would be most beneficial will guide government convergence. At the same time, the informal but strong networking which characterizes the OSS movement and its most successful projects should also be deployed to ensure knowledge sharing and active communication between interested stakeholders.

CONCLUSION

The TTLUG is in favour of the government conceptualizing and implementing a useful FLOSS policy to benefit Trinidad dnd Tobago and its people and enthusiastically endorses such a strategy being adopted by Trinidad and Tobago. We believe that FLOSS works both as a desktop and a server platform, and that the country can realize significant benefits, flexibility, cost savings and implicit and non-measureable benefits by adopting a FLOSS-favourable strategy. The discussion paper put out by the MPAI, despite its many flaws, is a useful first step on the road to a future with FLOSS and proprietary software coexisting and interoperating together.

However, we also recognize that FLOSS, as good as it is is, is ultimately a tool. Efficient process demand that the best tool should be used for the job, and FLOSS should only be used in cases where it is the best, most effective or least cost option. However, the shift from an almost exclusively proprietary system to a system with OSS as a vibrant and valued part of the national IT ecosystem requires a comprehensive long-term plan. Ultimately, we are still concerned that any pro-FLOSS strategy may be overridden and sabotaged by those with a vested interest in the status quo, and that public viewpoints will not be considered by those with vested interest. The existing lack of transparency in IT contracts and the exclusive use of proprietary system means that there are openings for unofficial considerations being brought to bear. The adoption of FLOSS software in concert with a more transparent procurement process means that most of that "fat" in the system is skimmed off. While this benefits the country, it may not benefit those who profited under the proprietary system.

The implementation of a FLOSS-favorable, or even a proper FLOSS-neutral policy requires long-term commitment, and success will not be instantaneous. At the same time, delays in implementation schedules and other temporary setbacks should not be used as an excuse to return to the status quo of proprietary

software and software and data lock-in. The implementors of the strategy must do their research and be willing to deflate unrealistic expectations. With regard to end users, some training will be required to get them up to speed with OSS applications, but the beauty of OSS is that it is available in an almost infinite variety of interfaces, which means that web-based interfaces (now possible with AJAX and Web 2.0 applications) and interfaces that closely emulate familiar metaphors will be available to end users, to name two. In contrast, a change in interface of proprietary software (as is expected in Microsoft Windows Vista, available in 2007) will also require significant retraining, as well as the increased license fees, greater hardware expenditures and reduced functionality (application and macro breakage, etc) that new releases of proprietary software seem to bring.

On the government side, the government should promote education policies that focus on concepts,and not just applications. The use of neutral policies mean that training can provide knowledge for both proprietary and OSS , reducing retraining costs in the long run. This educational thrust will benefit the country by equipping its people with more portable skills, which are not tied to a particular architecture or application. The State must also not be afraid to draw upon the vast body of international experience and case studies with regard to FOSS implementations (both successes and failures are learning experiences), both at the public sector and at the national level. However, since every experience and circumstance is unique, the State must also be willing to adapt and mutate its policy as needed and as circumstances dictate.

FLOSS software is not perfect. There are issues associated with FOSS software, some with the potential to be serious. To name one issue, patent and copyright issues could potentially be used to impede open source projects by those with a vested interest in a proprietary status quo, but this is more a function of a badly broken system patent system than with any deficiency of the software. In fact, the same patent issues that bedevil OSS also affect proprietary software and

unlike proprietary software, identified infringements will be quickly coded around. However, unlike other types of software, FOSS has as one of its greatest strengths, a strong, viral and vibrant community which is self-governing around merit and professional pride (the best ideas rise to the top and in the long term, the best people do so as well) which constantly identify and aggresively solve said problems. In fact, the open and frank self-assessment and self-criticism might lead the uninitiated to believe that the harshest critic of FOSS is the FOSS community itself. However, it is **precisely** this constant self-criticism which has improved FOSS and the FOSS ecosystem to the point where it is having such a world-wide impact to the point where it is now seen as the leading innovation in software development for the past thirty years.