ECONOMIC DEVELOPMENT AND INFRASTRUCTURE COMMITTEE

Inquiry into Improving Access to Victorian Public Sector Information and Data

Canberra — 13 August 2008

Members

Ms C. Campbell	Mr D. Davis
Mr P. Crisp	Mr B. Tee

Chair: Ms C. Campbell
Deputy Chair: Mr D. Davis

Staff

Executive Officer: Dr V. Koops
Research Officer: Ms Y. Simmonds

Witnesses

Mr A. Chatterjee, Manager, OzLabs, and

Mr P. Russell, Computer Programmer, IBM Linux Technology Centre.

The CHAIR — Good morning, gentlemen; I have had the opportunity to welcome you and introduce you to the committee members. This is an all-party parliamentary committee and it is hearing evidence today on the Inquiry into Improving Access to Victorian Public Sector Information and Data. All evidence taken at this hearing is protected by parliamentary privilege. Comments you make outside the hearing are not afforded such privilege. Could I ask each of you to please state your name, your business address and whether you are attending in a private capacity or representing an organisation, and if you are representing an organisation, what position in that organisation. Thank you.

Mr CHATTERJEE — My name is Abhisek Chatterjee, and I am representing IBM. My role is the Manager for OzLabs, which is part of the Linux Technology Centre. The Linux Technology Centre is a global organisation for IBM that does a lot of work with open source software and is basically the centre of excellence for Linux in Australia.

The CHAIR — Thank you. Could you give your business address?

Mr CHATTERJEE — The business address is 8 Brisbane Avenue, Barton, ACT 2600.

Mr RUSSELL — It saves me doing it. I am Paul — Rusty — Russell; I am one of the developers at the IBM Linux Technology Centre, which means I am actually a computer programmer. My official title within IBM is Kernel Hacker.

The CHAIR — Did you say 'official' title?

Mr RUSSELL — Yes.

The CHAIR — Okay.

Mr RUSSELL — Except on my original business cards there was a typo and they said 'Kernel Hocker'. I still keep those around for special occasions. I work at IBM as well and I am here to represent them. It is the same address; we have not moved.

The CHAIR — Thank you. I understand you will give us a brief presentation and give us the opportunity for questions. Over to you, Rusty.

Mr RUSSELL — Brief was the operative word, I think. My speciality is open source software; that is what I do for a living. Generally I will be focusing on the later questions, but I thought I would throw some things out there and then let you inquire. I was also the intellectual property policy adviser for Linux Australia a few years back. I have since stopped doing that task, but I still have a good relationship with Linux Australia as an organisation. In that sense I do have an interest in the rest of the questions, but I will be focusing mainly on the open source question, which is basically from question 18 onwards.

Particularly I want to focus on question 20: what is the capacity for both software models — being propriety and open source — to coexist in the same organisation? This is a really important one to address at the beginning. There is a saying in software: all software sucks, some just sucks less than others — and as a practitioner I can assure you that to some extent it is true. When we talk about open source, proprietary software and everything else we are really talking about the licensing; we are not talking about the software itself. There is poor proprietary software and there is poor open source software. It is important to distinguish between the licensing of software and the software itself. You could find examples on both sides of whether it is a superior product, one side or the other. But fundamentally if a vendor offers you software with a more generous licence than another vendor, that really is not your concern. As somebody who is interested in the software itself and in using it, whether it is open source or proprietary does not really matter. It is a licensing question. Now the fact that the licensing is more generous and it allows you to do things that you probably would not be able to do with the proprietary software opens new opportunities and you feel free to explore those. But you have to understand that fundamentally you can treat the

two as identical things, if you wish. There is no additional burden on you simply because the person providing the software has chosen to give you extra rights, provides a more liberal licence than is perhaps normal in the proprietary world.

So questions like, 'What risks?', 'What opportunities?' and, 'Can they coexist?', are somewhat misguided. In the same way that you can have, obviously, solutions and do have solutions from multiple vendors in an organisation, you can have solutions with multiple licences. In fact even with proprietary software there will be variations in licensing terms. One advantage that you could put is that, given that the licensing terms tend to be more liberal, they are harder to violate accidentally. To questions like, 'Tracking licences?', if one of the employees likes the software and decides to take a copy home, these are not problems if it is an open source product, by definition.

Where we do sometimes get into a legal quagmire is when you do take advantage of some of the extra rights; you decide that you should start distributing the software yourself. And this is not something that you can usually do under a proprietary licence. If you purchase a copy of Microsoft Windows you probably cannot sell copies on to other people; you probably should not give them away on street corners. Now you can do that with open source software. At that stage we are talking about licensing and questions of, 'What do those licences require of you?'. And they vary from requiring very little other than attribution, to requiring nothing at all from you and you may do whatever you wish with the software, to placing obligations that if you pass the software on to someone else they get all the pieces that you received in the first place. But these licensing issues only kick in should you decide to go down that path. And there will be cases when you do. But in general, as a question of procuring software, looking at solutions and things like that, these issues certainly do not affect you.

On question 21, about the role of the Victorian Government in procuring and distributing open source software, the procuring I think we have already covered. There will be cases when what you will be procuring is an enhancement to existing software. That is one of the strengths that we have seen in open source software, that obviously since anyone can enhance it, people do. Frequently there might be a software product that is close to what you need but not quite there. There is quite a booming trade in Australia of companies that will produce boutique solutions, modify software and enhance it in various ways. In that case the licence of the software product may or may not require that that modification itself be open source. If it does, then it is fairly simple. It is up to the vendor to make sure that they follow the rules there. If they do not then it is just like procuring any sort of boutique software, the licensing terms are part of the terms of sale, whether they keep the copyright or the copyright gets transferred.

There is an argument that if development is being done on behalf of a government agency by a third party, that if they wish to lease it under open source licence, then there is less concern over who has the copyright because you get so many rights that are normally withheld by the copyright owner when it is licensed that way. The question of who technically owns the copyright licence is much less important because even the non-copyright owner can distribute copies, they can make modifications and do all those things. Generally the difference is only important if you try to enforce the licence for some reason if someone does manage to violate the terms somehow; the copyright owner tends to be the one who has to do that enforcement. That is extremely unusual, so in practice it is less important.

With proprietary software, who owns the copyright is critical because they are the only ones who have the right to distribute it, to modify it, create future versions and often they are the only ones who can actually fix bugs. You can certainly argue that if a vendor wishes to retain the copyright on something they have produced particularly for government, they should pay for that privilege.

The CHAIR — Sorry, could you say that again?

Mr RUSSELL — If there is software produced particularly for government where they do not wish to release it under an open source licence, and they wish to retain the copyright, the government is funding the work, but they wish to retain the copyright so they can sell it on, that is some extra value that they have obtained. Presumably if you are funding developments so that they can then go on and sell it to other people, then they should be taking some of that responsibility since they are getting the benefit.

Mr TEE — It is the same if government gets research done, who owns the copyright in terms of the product?

Mr RUSSELL — Who owns the patents and things like that — exactly! Now if they are going to end up with it then there is an argument that says that they should pay some amount for it.

Mr TEE — Or we should pay less for it.

Mr RUSSELL — You should pay less for it — exactly! It is some value. In the case where they are open sourcing it, then it will be available for all and it is much less of a question, so you could argue that if they are doing proprietary development for you and they are going to keep those rights, you should be paying them less.

There are a couple of second order effects that we see when software is released under a open source licence. There are two particularly from my experience that I want to point out. The first one, which is not immediately obvious, is talent retention. In my experience we have seen an improvement both in the morale of people working on open source software and in the quality of people that we can attract to do such development. The reasoning is that generally if you have an in-house solution, you have had it developed and it is software that only you use, and if they spent five years working on that, it is a dot-point on their CV, but it is not really something they can take with them. If you are developing an open source solution, firstly other employers can see what they contributed directly in a lot of cases, also it is quite possible that other people are using this software, so the kudos and the ability to carry the skills with them are very valuable for an employee.

The flipside to that is that kudos is often lacking when you are working on a small team and nobody else knows really what you are doing, and it can be a great benefit to somebody who is coming through the industry and it can be quite motivating. Certainly we have seen that employees who start working in open source software find it very difficult to go back to working on proprietary software because they really do enjoy the kind of feedback they get and the exposure they get. Not everyone becomes an open source rock star but there is certainly an element of exposure that they enjoy.

The CHAIR — Can I be absolutely clear on what you are saying there, that by implication in the past those higher up the line in management tended to take the glory for good products, and this is one way that a rising star can be recognised in their own right that gives them job satisfaction and obviously also the ability to move up the career path fairly quickly within that organisation, or be recognised outside, and therefore even if their own particular company does not want to recognise them for good HR practices, they do it for retention of talent. Is that what you are saying?

Mr RUSSELL — That is certainly one corollary. The other thing is that there are a lot of talents in IT that simply go unrecognised. They do not necessarily give anyone credit up the chain because they are deeply technical, they might be neat for techies but they are not really the kind of thing that gets reported upwards, and they do get exposure for those as well because other technical people in the open source world will see what they are doing and sometimes be quite impressed even though it is not the kind of thing that really turns out to be a bullet point in a report at a higher level. The other point is that those employees will often then be able to talk about their

work and have exposure to their work, which attracts other people. When you find someone doing exciting work like this in an area it attracts graduates, it attracts talent to that organisation and we have certainly seen that.

The other point I would like to make, it has certainly been the experience within IBM, is that in a number of cases theoretically there has been nothing stopping us sharing internal software between one area of the organisation and another. We are all IBM, right? So across the organisation it should be possible. In practice, however, there have been a number of cases where the simplest solution has been to get one department to open source the software and the other department to get it from outside, effectively circumventing the siloing effect that tends to inevitably occur. I am not quite sure why this happens. It certainly should not, but by exposing the software I think we see two effects. The one is reduction in the amount of turf wars and fiefdoms that we see. The second is that the communication then tends to occur at a lower level; it tends to be technical people who are looking for a solution who go out and find these things elsewhere, trial them, once they have decided that they are sufficient, then they look higher and try to get approval for perhaps a more significant test.

When you are in that situation and you are looking for something, you are not likely to go through all the sometimes convoluted processes even if you know that a solution possibly exists in another area, to get approval to perhaps get a trial version and all those things. In a lot of cases you are not quite sure that it will meet your needs anyway, so if somebody has got a problem in front of them and is looking for a solution, it is far easier to go and find some open source software, try it and if it does not work, they have lost half an hour. If it does work then it is time to take it further. Sometimes just having to ask permission can be a significant barrier.

The CHAIR — Just because they ask or because of the elongation required to get approval?

Mr RUSSELL — Partially because of the elongation but partially just because they have to ask if it is an idle curiosity. Sometimes as a technical person if they think, 'There has got to be a solution to this that is fairly simple' and they are not quite sure what they are looking for: it is not the kind of case where if you do an internet search and you find some open source software out there, you talk to a couple of people and they say, ' I think that will work', you can try that, you do not have ask anyone, you are not putting yourself out there quite as much as if you had to actually talk to the person who provided it. Even if they quite willing in response to a phone call to talk to you about it and give you a trial, you are not at that level of commitment. A lot of people like the ability to try things and if it fails, just bury it silently.

Mr CRISP — Because it is allowing access to experimentation that leads to innovation if you do it quickly.

Mr RUSSELL — And the vast majority of those cases, will fail. You know that, that is something that might not work, it probably will not work, so it is not really worth bothering somebody about it, given that it is an off-chance.

Mr CRISP — But it is those off chances that lead to innovation.

Mr RUSSELL — Sturgeon's law — 90 per cent of everything is crap. This is very true when we talk about almost anything but certainly about the release of information. The vast majority of it is a failed experiment and will not be useful to anyone. The problem is that in the extreme case Sturgeon's law is believed to be irreducible: even if you try to identify the 10 per cent that is not crap you will still end up with 90 per cent crap and vice versa. Trying to pick which bits are going to win is often just more effort than simply saying, 'Let somebody else do the dirty work and figure out what is the worthwhile part'.

The CHAIR — Is that your presentation, Mr Russell?

Mr RUSSELL — Yes.

Mr CHATTERJEE — Just adding my 2 cents worth to what Rusty has said, I had a lot of experience in the Commonwealth Government prior to joining IBM and also in the state government where I worked across a lot of projects. There are a lot of questions about open source versus proprietary. I have seen, through my experience, that open source has worked really well because of the access to the software and because of the access to the talent and the number of people who are contributing to the open source. A lot of people worry about something, which is called maturity for open source software, whether it is capable enough, unlike an off-the-shelf product, and a lot of people worry about support — if we put it as a business-critical system whether we will get enough people to support it and so on. But things have changed. Previously open source was the edge of the network, as we call it, where people were just testing them out. Now I think the time has changed. We can see more and more systems, which are business critical and are all based on open source and Linux. If you look at the top 500 supercomputers that are there on the market, everything is based on Linux. Because of the fact that we can get more out of the system, it tends to fail less and it is more efficient. A classic example of that is the IBM Roadrunner, which we have just released on the market and, again, it is based on open source. It is the fastest supercomputer and does 1 petaflop processing per second. It is amazing to see those results. In Australia we have got a very strong base with some extremely talented people. Our mission is to grow and be the evangelist for open source at Linux and to help it grow better in the market, to create more awareness amongst the community to make people aware that we are here to help them with those sorts of things. The capability is huge; it is amazing.

The CHAIR — Before we ask questions I think it is important that reference is made to Linux in Hansard, because this will go on the internet. We have got a very good briefing paper, which I presume you have supplied, on Linux.

Ms SIMMONDS — It is from the Web.

The CHAIR — Could you succinctly summarise what Linux is, so that people who are reading our website know what you are referring to?

Mr RUSSELL — Sure. Linux is what we call an operating system. It is a piece of software that runs a computer on top of which other software runs. You might be familiar with Microsoft Windows, which is Microsoft's operating system; Linux is an alternative. Linux was developed and started in 1991 by a student called Linus Torvalds, after whom it was named, and is now developed by a worldwide group of engineers, including myself. But I think you really have to understand the scale of this thing. IBM sells billions of dollars worth of hardware every year. We sell low-end simple machines, such as PCs that you would all be familiar with, that are running Windows. We sell middle-range machines — the $100 000-ish range up to about $1 million; that is what we call middle — generally running AIX. Then we have a high-end mainframe sold into financial institutions and banks — people who cannot afford any kind of failure — and that runs a system called z/OS. IBM has run this way for years.

There was some speculation that it would be possible to have one operating system, instead of these three different ones, that ran across from our tiny embedded machines that sit inside devices all the way up to the high-end machines, but we are only a multinational and did not have the resources. It is not that it was impossible; it was simply impractical. It was considered not something that we could actually do. There were a few attempts at it, but you just add up the numbers and you go, 'Nobody can afford to build such a thing; it will almost certainly fail'.

Linux is that thing for us. It was developed externally, it was ported originally to the standard PCs that everyone has, but it grew from that. A group in Germany, which works on our very high-end machines made it run on those machines — it was a bit of a technical bravado kind of exercise — and then customers started asking for it so we started selling it. Now up to half those extremely large machines sell with Linux on them; it is a multibillion-dollar business for us; it is the only

operating system that runs all the way from the very tiniest devices we make up to the very largest. We could not have built it by ourselves. We did some enhancement. In 1999 we committed $1 billion in development on the Linux platform and we have put in more since, but that is a drop in the ocean compared to the other work that everyone else has done. This is a massive infrastructure project that nobody is saying would have begun with the intent of making an operating system that could run everywhere. It simply is not feasible, but it has been done.

The CHAIR — Right, thank you. We will move to questions. In terms of our briefing papers, there is a reference made to balance of payments. It says that Australia loses more from ICT importation than it gains from wheat and coal exports. We are the Economic Development and Infrastructure Committee. Have you got any comment to make on your presentation about what we should be doing in order to address the economic issues as opposed to the technical issues?

Mr RUSSELL — I am doing the best I can.

The CHAIR — You are doing very well. It is very interesting.

Mr RUSSELL — There is a great deal of talent in Australia in ICT, but traditionally we have been something of a branch office economy in this area, hence the high rate of imports. One thing I find really interesting is that the Boston Consulting Group survey, done several years ago now, had a disproportionate number of Australian developers working on open source software. We have a great deal of strength in this area. I would argue, especially as somebody with a vested interest in working in this area, that that is something we should foster.

The CHAIR — How?

Mr RUSSELL — That is a very good question.

The CHAIR — Have you got any good ideas?

Mr RUSSELL — Interestingly the government is the largest procurer of ICT and at the very least ensuring that there are no artificial barriers in place. In some ways even today the proprietary software is still seen as the established, safe option and that open source is this radical, hippy idea that is a little bit risky. That perception is, I believe, not correct given the amount of professional development going into open source software. I think it is fairly clear that the software should be taken on its merits and not necessarily on a question of licensing. But further than that, this is an area where services rule. Because the licensing is permissive enough that others can take your software and sell it and be in competition with you, you really end up competing on the basis of services. Services is an area where Australians seem to have a great deal of expertise. IBM has 15 000 staff in Australia, and the vast majority of those are in services.

The CHAIR — If you were putting one or two recommendations in our report about ensuring we get the best out of our Australian talent and use it to export, would I take from what you have just said that government should give a significant focus to open source, no. 1? What would be your second one to ensure that young vibrant talent, or middle-aged vibrant talent, is harnessed?

Mr RUSSELL — Interesting about your ‘middle-aged’, because one of the surveys put the average open source age at about 34 or 35. Having grown up over the years, from sort of 10 years ago it was probably 10 years younger, we are not all spring chickens any more.

The CHAIR — I put middle-aged as about 54, not 34!

Mr CRISP — For Hansard, honourable older members interjecting.

Mr RUSSELL — For the ICT industry that is quite eye-opening for some people, because it tends to be perceived as a very young person. People come out of college and they start programming and they change the world. We are definitely seeing a shift. The bell curve is all the way up. I know open source hackers who are 60-plus. There does not seem to be a barrier because there is no barrier; it is a matter of interest and capability and anybody can get involved.

The CHAIR — Now that we have taken you off track, what is your second recommendation to make sure that this vibrant talent, regardless of age, is harnessed, no. 1, and of course the by-product of that would be it could increase and assist our balance of payments?

Mr TEE — Isn't the problem, surely, that it is just that we import a lot of hardware from Taiwan. You are never going to in a sense compete with that. That is a product of — —

Mr RUSSELL — The hardware keeps dropping in price. Every 18 months or so, it pretty much halves.

Mr TEE — But in terms of our balance of payments — —

Mr RUSSELL — It is the software that is messing us up. Software does not drop anywhere near as fast in price as hardware does. It tends to only go up somehow.

The CHAIR — We keep hearing that in government.

Mr RUSSELL — That is right.

The CHAIR — Your second recommendation?

Mr RUSSELL — Structure I think is important. People feel a lot more reassured if we are looking at enhancing the ability for people to utilise this stuff. We know a lot of it is out there. They have to find it, they have to have support for it, they have to know they are not colouring outside the lines if they decide to use it. This means that whereas previously you might have had a whole-of-government contract with a particular proprietary vendor you should look to make sure that those who decide to perhaps experiment with open source software get the support that they need, that they have somewhere to go.

Particularly we see this in education. A number of schools are looking at rolling out open source software. They basically have to go on their own if they decide to do it, and yet the budgetary cases for a lot of those are very compelling. I started programming before high school age and that is exactly the sort of time that if kids have access to this sort of technology they will do that. As someone once said, 'Why would you buy a car with the bonnet welded shut?'. Open source technology gives you all the pieces to, if you have the interest, go in there and find out how it works, and that can certainly lead to greater things.

Mr CRISP — I wanted to ask a couple of questions to crystallise a little more where governments are. Are Australian government leaders or followers in open source?

Mr RUSSELL — That is actually a very hard question to answer. There is some degree of the grass is always greener on the other side of the hill — you hear these great stories from other places of some things that they are doing. There are places that are definitely doing some great things, especially in places where, frankly, they have hit a budget wall and they just cannot afford it any more and have been forced to review options that are perceived as more risky. Especially if you look going back five years — Extremadura in Spain did a big rollout, Brazil has been quite aggressive in adopting open source software on a large scale.

I think to some extent we have been cushioned a bit because we do not have the same kind of pressures, but they are still there. We are certainly not at the back of the pack either. There have

been some very good government actions on this. AGIMO has — they are no longer called AGIMO, are they; they have a new name I think — —

There are open source software resources. I can give you some of the URLs for those. The finance department actually has quite a good open source resources section. Definitely this awareness is bubbling up throughout the government, certainly because our Australian-owned ICT sector here is dominated by small and medium players. They tend to have been quite aggressive because they have got similar kind of budget constraints. They will tie things together with shoestrings if they have to. They were often very early adopters of open source. There is actually a depth of expertise out there. That bleeds across into government as well.

Mr CRISP — It only bleeds across. I guess you have answered this question but I want to get it on the record: Is it right that governments are not training or leading the way, that bleed-across is from private sector to government in skills rather than the other way, that government is not providing any contribution in this area?

Mr RUSSELL — 'Any contribution' perhaps is a bit harsh but certainly I would say that most of the flow would be the other way. Where governments have provided things they tend to be in reassurance — reports, summaries, guides, things like that — that it produces quite well on open source, which has definitely been positive. As far as actually training people up and being a thought leader in that area though, not so much.

Mr CRISP — Thank you.

Mr TEE — One of the areas that I suppose we are looking at is whether or not the Victorian government should be more open to use of open source material, and I suppose that comes back to the issue of the risk. You have identified the risk around maturity. I suppose what you are saying is that now the industry has developed to such an extent that that risk is a hangover of the past rather than a reality of today. I suppose I am just trying to identify what some of the other things and issues are that we ought to be aware of. If you buy a proprietary model, do you get a bit more certainty in the sense that the owner of the product is the one that you are dealing with — —

Mr RUSSELL — Until they go bust, yes.

Mr TEE — Yes, although you could buy from a reputable owner — —

Mr RUSSELL — And they never go bust! Or end-of-life products.

Mr TEE — The other issue is the complexity issue — that yes you can have a coexistence but does that just mean you have got a greater complexity? What are the risks and what are the answers to those risks, and really I suppose what are the advantages? Are you buying a system that is going to keep evolving and you are going to keep being the guinea pigs?

Mr RUSSELL — That is a very good question. I think that applies with any software purchase, proprietary or otherwise: is it well-established? Are the people you are getting it from particularly reliable? Do they give good support? These questions are independent of the licensing of the software. When you talk about heterogeneous systems being more complicated, almost always that is true. Unfortunately nobody runs homogeneous systems because those systems are always too complicated. The exposure of vendor lock-in in proprietary software is very real, although you might say that this company is unlikely to vanish — and there are not actually many Australian-owned companies that you could point out and make that claim of in the ICT industry. Even the big ones vanish. But also there are the end-of-life products, or they decide they are going to change the way the next version works, and that one feature that you relied on in your critical infrastructure is gone. There is no negotiation; it is over, and you have to scramble and find something. Unfortunately you hit these issues. In an open source product you could then make the choice. If you really want that, are you prepared to take on the maintenance burden? You have

more options. They might not be options that you choose to take, but I find it difficult to believe that the existence of more options than you had previously is a net disadvantage.

Mr CHATTERJEE — The other thing to quickly add to that is that it is very important to keep in mind that IT is the enabling part and business is the driving part. It is very important to keep focus on what the business or government or whatever is trying to do and align your strategies with IT. A lot of people think to just reverse the strategy and say IT is the driver and business is the enabler but that is not right. From previous experience I think aligning the strategies and the business requirements would gain more value out of your IT systems. Then you do the appropriate planning to make sure that your open sourcing and proprietary softwares can coexist. Those things will come along provided we have got the right focus and the right plan in place. Otherwise things tend to be more expensive in the end. From my experience that is very important.

Mr TEE — All right. Just in terms of the bigger picture in terms of the proprietary versus the open source, is there a change in the market — is one winning over the other or is it a fight to the death or is it a coexistence that will continue? Where is the debate up to?

Mr RUSSELL — That is actually a very good question. The debate is still raging. There are certain market segments where open source is now dominant. There are other market segments where proprietary software is dominant, and the vast majority are in the middle, where there are both options. Interestingly, the fact that open source software tends to be cheaper — because it is widely available and if someone is selling it too expensively you can always get it from someone else — is not the driving factor. Nor it seems is necessarily directly the ability to modify but it is the fact that you could, or somebody could, that tends to level the playing field and provide that awareness that even if you do not require modification today, that is a door that is there if you need it and that if you are not happy with the support you are getting, there is absolutely no advantage that they inherently have over any other party from whom you could get support.

Mr TEE — But then if cost is not the advantage and adaptability is not the advantage, why is it that proprietary is not dying out?

Mr RUSSELL — Firstly, let me clarify. The direct adaptability is not always the advantage. It is not necessarily that they look at something and go, 'That would be great but we want to make this change', but the fact that others have made those changes in the past and have made them available, and the fact that you avoid then the lock-in. Even though you may not be directly interested in making modifications, the fact that you can, the fact that others have and the fact that you can get support for a wide variety, which has led from that ability to modify even though you do not directly, again we come back to the car with the bonnet welded shut. Even if you are not a mechanic, there is still an advantage in having access to those things. Why is proprietary software not dying out?

Mr TEE — Why has it not gone the way of the Beta as opposed to VHS? What is the — —

Mr RUSSELL — I think mainly because it is just not that clear. Proprietary software is more of a pay as you go. If you ship one million units, you will get one million times whatever price you charge for the units — very straightforward economics. With open source software, if you ship one million units, you almost certainly will not get one million times N. You are having to fund development up-front because you are going to get some other advantage. In the case of IBM it is, 'We are selling services or we are selling hardware that we are enabling or making more powerful by doing these enhancements. We are gaining a reputation as being leaders in this field'. That is exactly the kind of thing that we gain, so there are different economics in play, and it is not a clear, 'This one will be $100 and this will be $10, therefore the $10 one will win'. Also fundamentally we are talking about software. Some software is better than others. Sometimes the

better one is proprietary and sometimes it is open source, and they keep evolving so it is not even a static balance.

The CHAIR — Could I just take up a point that I think Mr Chatterjee made that the government does onsell some of the proprietary software that it has purchased. Have you got examples of that, because I have been trying to think since that comment was made of examples, and I cannot think of anywhere the government has onsold.

Mr CHATTERJEE — Did I mention onselling proprietary or did I mention onselling open source?

The CHAIR — No, proprietary.

Mr DAVIS — There is a tax on it.

Mr CHATTERJEE — Yes, tax, but is that direct onselling? It is not direct selling, though; it is still distributing.

The CHAIR — Perhaps the comment was made in relation to why there would be advantages in proprietary software and they had the ability to onsell, but my recollection is I have not read anything that refers to that. Again, could I ask Mr Chatterjee, because of your experience in the federal and state government area, without naming any particular examples on your part, I can think of one where we had an Auditor-General's inquiry into the Parliament — its IT system and so on, and its usefulness et cetera, et cetera, et cetera. In government what tends to happen is that when things go wrong, they go spectacularly wrong and they are very public. Have you got examples of where open source work has enabled a solution to be found far more quickly and therefore the risk to government is less because more people have become involved in solving problems that would otherwise just be left to those in the company that the government has purchased the product from? For me I see this is a huge advantage for governments and public servants that are naturally risk averse. Whoever is in government, there is always an opposition, and when things go wrong with IT, it is huge in terms of dollars.

Mr CHATTERJEE — There are a couple of things to remember with open source. As Rusty has pointed out, it can initially be looked at as a cheaper option, because there is no cost associated with acquiring the software, but if you look at the total cost of ownership — that is a term that we use — it is how much will the solution end up costing in five years time, and that includes maintenance, which is the most expensive side of any IT system because the actual software or the actual server, the actual hardware is nothing — about 10 per cent of the total cost. There are a couple of examples that I can cite, but unfortunately I cannot name the department because there were security clearance issues.

The CHAIR — I would not want you to, but are there examples of where open source has been able to work — —

Mr CHATTERJEE — Absolutely.

The CHAIR — Whereas if it had been proprietary it would have been much longer?

Mr CHATTERJEE — Yes, absolutely.

The CHAIR — That is fair enough.

Mr CHATTERJEE — There are more than four or five issues or cases. If you engage any proprietary software it comes with a very high cost to start with. It depends on the actual solution that you are looking at, and it also depends on the infrastructure that is already there, and then you have to strike a balance to see which one is the better choice. Because the place that I was in had a team that had some overview of what was happening, and we had in-house expertise,

to take on some of the existing work from the open source community and to add into our systems, and that worked.

The CHAIR — Thank you; that has been extremely helpful.

Mr RUSSELL — I did want to venture a little bit into the content side for a moment because I have a personal interest. Very briefly, two issues pervaded your discussion paper a little bit. One was the question of how do we decide what to release, and I think I already made the Sturgeon's law reference there. I think it is theoretically possible to establish which bits are important and which bits are not, but the cost involved in that is probably greater than the benefit that you are going to see anyway. The other pervasive issue, and the one that we see quite a lot in the open source world, and we wrestle with this, is what if somebody goes out and takes what we have done and gets rich off it basically? This happens quite frequently in the open source world. There are a number of places that make a significant amount of money selling the software that I have written without asking my permission, and that is perfectly allowed, and this is exactly the same thing. Whenever you release something under content you might discover that somebody goes and makes $1 billion off it or a large multinational like Google takes it and does something that turns out to be extremely valuable to them and earns them significant ad revenue, which is basically how they make money. Really this comes down to an attitudinal question more than anything else. If your neighbour wins the lottery, you really have two ways you can take it. The first way is to realise that by winning the lottery they reduce your chances of winning, because if their ticket had not been chosen, your ticket was more likely to be chosen.

Mr DAVIS — But not much.

Mr RUSSELL — By some world view they stole the lottery. The other way is to wish them luck and hope to borrow the Ferrari on weekends, and I am very much the borrow the Ferrari on weekends-kind of guy, especially since I do not buy lottery tickets, I was not going to win, and that is exactly the point. You do not want to be in the business of buying lottery tickets. For everyone who comes out there and does great things and they earn a whole heap of money and somebody goes, 'But that was on the information that you released freely', there are 99 others who tried it and failed. You do not want to be in that business. Let them go out, and occasionally there will be a success, and that is great. You will either get it back in terms of tax revenue or at the very least by making productivity gains for Victorians because the data that has come out of the Victorian Government is almost certainly by its nature interesting and useful to Victorians, so as a tendency, even if you do not get direct tax revenues because it is not necessarily a Victorian organisation that ends up using it, you do get benefits for all Victorians because that information has become more useful to them. I think that is the kind of viewpoint that I attend with these things, and if somebody strikes it rich, more power to them.

The CHAIR — Thank you very much to both of you. Within a fortnight you will be provided with a copy of the transcript. You are familiar with the system. You are allowed to make typographical corrections but not change the content. Thank you very much, and the world will have the benefit of your evidence when it goes up on the internet.

Witnesses withdrew.