



WHITE

The “Blue, White, Red” technology trilogy
By Marc Fleury

**“How I learned to stop worrying and love the
business of Free Software”**

Copyright 2003, The JBoss Group, all rights reserved

WHITE

Second part of the Blue, White, Red technology trilogy

“How I learned to stop worrying and love the business of Free Software”

The question I am about to address has been best summarized in an article recently published by Business Week March 3, 2003. *“A ragtag band of software geeks is threatening the established software players.”* Indeed, how can thirty guys around the world, the size of the JBoss development team in first quarter 2003, compete successfully with the likes of IBM, BEA and Microsoft?

The answer is simple. We are all witnessing, first hand and without necessarily being conscious of it, the virtuous circle of the Net and modern Free Software. *“Open source Software is as large and powerful a wave as the Internet was”* says David Stutz, former GPM for Technical Strategy at Microsoft. We agree, they are one and the same. Open Source made the Net Era possible. The whole Net almost exclusively runs on Open Source/Free Software. In return, the explosion of the Net nourishes and distributes Free Software development. The Net not only impacts technology development; it profoundly impacts the technology business.

The Net and Open Source make it possible to produce software cheaply and distribute it at zero cost. The larger macro-economic trend of commoditized infrastructure software is here to stay. JBoss and JBoss Group did not create this trend. The story of our business is a

micro-economic story of how we came to understand and leverage this larger phenomenon. The Net is weaved in our business model from the get go and it impacts the way we do business every day. Gone are the days where the best way to establish client-supplier bonding was over a game of golf.

Sales-Engineer

I must admit that I am having fun selling Free Software. I haven't had fun in Sales since I started selling for Sun Microsystems France. I am forever grateful to them for hiring me in the beginning. I had just finished my Ph.D. and had no real-world experience, except for the long hours I had put in my Physics and Mathematics degrees. Microsoft rejected me when I was looking for a job, with a form letter explaining that in the face of the tens of thousands of applications, mine didn't really stick out at all. Yeah, I can see that. The HR director for Sun France described, almost apologetically, the unglamorous job they were interviewing me for: “Presales phone support agent,” meaning it was a mild Sun version of telemarketing. I was so naïve, I thought I was king of the world when I actually got the job. It was called Sun Info-Tech. And then the job actually turned out to be the best one I had at Sun. Although I wasn't commissioned, I remember the adrenaline rush of that \$60k fax coming in like it was yesterday. It's the kind of instant gratification that very few disciplines outside of pure sales or pure engineering provide you. Next thing I knew I was evangelizing Java to large crowds as the official spokesman for Java in France. After sales at Sun I decided that I wanted to delve back into engineering. This was greeted by some

skepticism by my colleagues, one who told me “You’re moving down the food chain,” emphasis on “down.”

Build it and they will come

Before there was a vision for the business of JBoss, there was the product. In my previous paper, “Blue,” I talked about the desire to build JBoss and its technological underpinnings. While the product was greeted with enthusiasm from the outset and quickly experienced rapid growth, finding the business for JBoss took longer and was subject to missteps along the way. A previous company wasted one year trying to come up with a business plan for JBoss. The elation of getting an interview with Sequoia Capital quickly dissipated as Doug Leone told us “This is not just a bad business plan, it’s a horrible business plan.” After the failure of that first company and accompanying loss of my savings, I had no choice but to leave Silicon Valley, pack up and go home, but not even to my or my parents’ home. My wife, daughter, dog and I moved to Atlanta, GA, and lived with my in-laws for six months. I was still convinced JBoss would take over the world and I remember arguing passionately with my lawyer, Larry Rosen, that we were the “original” garage company. Larry dryly responded, “Marc, you’re not a garage company, you’re an ‘in-laws’ garage company.”

While the business plan of the previous unsuccessful company strayed far from our core competency, my first clients brought me back to my area of expertise: JBoss knowledge. They were early users of JBoss who found my address on the Net and contacted me for training. On the success of those first onsite trainings,

I decided to give it a go at setting up an open-enrollment training. The very first onsite training sold-out, and I was on my way to making a good living doing what I enjoyed. The next step was transforming a personal consultancy into a successful services company.

Finding the Right People

After one painful experience, I was not about to lose my savings twice. I vowed that the next company I ran would be profitable and self-financing or it would be a no-go. This meant slower growth and only taking on people as the business justified it. Not having money is actually a plus when it comes to some kinds of recruiting. You weed out all the wankers and mediocrities who parasite off successful big companies and funded start-ups.

All the JBoss Group consultants started out working for free on open source JBoss. One of them stood out from the beginning: Scott Stark. While I was just a regular sales engineer at Sun, Scott had actually made it to the top of the IT hierarchy and pay scale as a lead architect with Bear Stearns. He had more experience than I did and fewer illusions. I tried to sell him on the radiant future of JBoss Group and he told me to shut up, that it was enough to do work he enjoyed and all he asked for monetarily was enough to cover his living expenses. He took over JBoss development, coded the much-needed security framework, and gave us an industry-leading implementation. Scott took a prototype and made a real product out of JBoss. This is where we started seeing mass acceptance of our product. I am proud to have Scott as my first business partner and proud that last year

he made more money with JBoss Group than he used to make at Bear Stearns.

The people who make it at JBoss Group display not only superior technical talent, but also business skills, risk tolerance and the ability to work with others. One of the most frequent criticisms of the JBoss 2.0 series was the lack of clustering. Bill Burke and Sacha Labourey volunteered to take on the task. Bill was a lead developer at Iona on their Orbix2000 team. From the open source JBoss work and independent consulting assignments with the company, he made the full-time commitment to join us. Today, Bill is the lead architect in charge of putting out JBoss 4.0.

Scaling up to meet demand, we recently opened an official Europe office in Neuchatel Switzerland and Sacha Labourey agreed to head our European operation. There is a tremendous opportunity in Europe but the JBoss Group is mostly perceived as a US entity. That is inaccurate since at least half our core developers are based in Europe. Sacha is a prototypical JBoss Group member, who combines software and business savvy. Before joining JBoss he ran his own successful independent consultancy. Bill and Sacha are the kind of talent that have built JBoss and will drive it going forward.

There are a lot of people who talk about open source who don't work in this field. One of the biggest illusions these people promote is the "open source community," the idea that once a product is launched into open source, a vast amorphous community will start working on it. This is only true of a few successful, large-scale open source

products and it is mainly true at the QA (quality assurance) level--the people who submit patches and bug fixes. Once you do reach a large user base, the QA community is a powerful resource to draw on and is a major reason why successful OS projects are known to be extremely stable. JBoss is an extremely popular product and our QA community numbers in the thousands. However, in the case of the actual development, the number of contributors with read-write password, meaning they can directly commit their code, has numbered 82. At any given time, we have about 10 people working full time on the container. Translation: we know exactly what is going into the product and who is contributing it because this comes from just a few people who work together full-time. JBoss development is done in an organized, modular fashion. JBoss Group manages the core developers, setting the overall goals for each sub-project, distributing the responsibilities and reviewing the commits.

In the beginning there was the Brand

"You have a brand", this is how one of the major analysts in the middleware field welcomed us. Brand is one of the few things you can sell in the open source realm. Anybody can take JBoss and call it Joe's application server, but what people want and trust is the JBoss brand. Our software is free; our brand is not.

It would be nice to say that I am a sales genius, but the truth is that for the first year of JBoss Group's business, I never once picked up the phone. All business came to us through our website. It would also be nice to say that the JBoss website

has been a marvel of communication and PR. While time and resources are now allowing us to focus more on sales, marketing and public relations, what purely drove our business in the past and what we will reinforce going forward is the JBoss brand.

This brand is based on the quality of our product, which is really excellent. I know, I know, we all say it, all vendors say it. But really, as a techie talking to techies, let me repeat, our product is excellent. Don't just take my word for it. JBoss won the JavaWorld Editors'Choice award for "Best Java Application Server" in 2002, beating both BEA and IBM. Our competition desperately tries to position us as competing with their low-end offerings. *"WebLogic and projects like Tomcat and JBoss serve different parts of the Java community"* says Scott Dietzen, CTO of BEA, on dev2dev in march 2003. You get what you pay for, right? Wrong. JBoss is not just a high-volume, low-end play, although we do dominate that sphere. The reality is that in many cases we lead the market with groundbreaking features our competition ends up copying, such as JMX, JAAS integration, hot-deploy, microkernel, AOP architecture. These high-end features position us as a formidable competitor to their high end products. Our push is clearly ubiquity, domination across all spheres. Trying to pigeon-hole us in the low end is a mistake.

How Open Source Works

How does Open Source work and how does it not work? Two myths: open source works for all software and it has

zero development costs. This brings you back to common sense. Developers enjoy and build reputations on their ability to write abstract infrastructure, system-level software. In contrast, end-user GUIs and company-specific applications are what waiting tables is to an aspiring actor. It's the day job. At night Clark Kent likes to transform himself into Superman and develop 21st century system-level code. This brings us to issue two. How do you transform your night hobby into your day job? This is where "Professional" open source comes in.

Not surprisingly, the answer is money. Open source does not have zero development costs. It has an order of magnitude lower development costs because it leverages the Net. While it appears as "free" to large companies with big R&D budgets, it is close to zero but not zero. I still need to put food on the table for thirty guys who could go out tomorrow and get highly paid, boring IT jobs. They would rather be working in open source, but they still have to pay the bills.

JBoss Group only pays people who have already proven themselves as JBoss developers and we pay for professional services. We don't eat the development costs up front and we don't pay to train unproductive people. We have a policy of allocating 50% of our developers' time to coding JBoss and 50% of their time to professional services. As far as I know, this is unique in the industry. It means that the open source development is led by real-life industry demands and, in return, our services come directly from the people who write the JBoss code and develop the open source knowledge in the first place.

This means you no longer need massive hiring and R&D budgets or a prime location in Silicon Valley to write any software. While it is easier and, in some cases, quicker to pay for software R&D up front, when it comes to infrastructure, successful Open Source will always catch up and surpass the quality and stability of proprietary alternatives.

Innovation and evolutionary dead-ends

Let me work some more on the fallacy that JBoss/Free Software lags in quality vs. commercial software. Strategically our competition has taken the strategic decision not to invest any more R&D efforts on their container. Their ROI on the core container is endangered by the advance of JBoss. That explains their desperate push to bundle web-services and integration and portals. They can't afford to massively invest in the container any more. We commoditized that field in effect in early 2002 with the release of JBoss 3.0, a state-of-the-art EJB container with full clustering.

First, our competitors ignored us, and then when we started to impact their bottom line, they changed their tactics to spreading FUD (fear, uncertainty and doubt) about JBoss. Their third approach is to allocate most their development resources to new, flavor-of-the-month features such as web services and portals. Key engineers in these companies that could advance the state of art in container design are instead pursuing fluffy buzzword compliance. What a waste of talent from my point of view.

It is obvious to me, as a technologist that we are in the infancy of container/middleware technology. Our containers are in Pampers. Don't get me wrong, JBoss may be the most advanced of all the container designs in the field, with research-grade innovation, but even that is recent technology. The best innovation, which I will discuss in the final paper of the trilogy RED, is still ahead with JBoss 4.0. We have irreversibly crippled our competition's capacity to develop their containers. They cannot justify, in the classic model of profit and loss, any significant investment in a commoditized piece of technology, which they know perfectly well we are distributing for free.

And the quality gap between JBoss and its competitors is widening. The fact that the research community is participating in our development validates this point. I am extremely proud of the fact that, among many high-quality academic papers, the paper we submitted with Professor Reverbel at Middleware 2003 was among the few accepted for presentation. The Middleware 2003 USENIX conference is the most prestigious academic middleware conference. It will take place in Rio de Janeiro in mid June and I am delighted to go there. Going forward, I am going to recruit more of the academic community in our effort. If I can achieve one degree of separation between MIT and Wall Street I will be proud. Academic research benefits from a focus on real-life needs, while the corporate community benefits from cutting-edge research. We are in the middle, making it real.

Now the business community doesn't just care about innovation. They care

about whether the product on which they standardize and the vendor who provides that product will be around for a while. This is a valid concern. Along with commoditization, consolidation is the other key trend affecting the middleware industry. Three years ago there were around thirty application server vendors. Today, there are less than ten proprietary vendors and only four of them have any significant market share. Even formidable IT powerhouses like HP have dropped the ball with their application server offering. Open Source is the only way you can ensure that you are not going with a product that will become an evolutionary dead-end, that is no longer advanced or even supported by its vendor.

JBoss becomes a standard

The Net gives successful open source software a massive distribution channel. If you have a quality product, this will build your brand. Bob Bickel, former General Manager of HP Middleware and, now, a JBoss Group Advisor, once said *“Modern technology needs millions of units shipped in order to be relevant at all.”* How true, I am not sure the drastic implications of this insight are obvious to everyone.

For example, ask yourselves, as venture capitalists do, “I need millions of units shipped in the field to be relevant. How will I reach this critical mass?” In fact, the cost of the marketing effort alone, not counting sales and distribution, is out of reach for all but the largest, established software players and Open Source. Even then, unless you are IBM, being a large company does not guarantee success; you also need a quality product. Many of our large

competitors have tried distributing their technology for free and marketing the hell out of it. Then they are shocked when people turn down the offer. As one blogger put it: free is too much to pay for a crappy product. Competing with free proprietary products used to scare the hell out of me; it doesn't anymore. I am told that VCs advise start-up companies against going into a domain that Microsoft is eyeing. The same advice ought to apply to companies seeking to compete with established open source infrastructure players. I just wouldn't go there.

“I am not too worried about JBoss, because it doesn't support many of the applications customers want to use” says Scott Handy, IBM's director of worldwide Linux solutions. The man is ignorant and essentially saying that customers don't want to use J2EE. Does he know what he is talking about? In 2002, we had 2 million JBoss downloads for the entire year. Already, in first quarter of 2003, JBoss downloads have almost reached 1 million. In contrast, the Sun J2EE reference implementation claims 2 million downloads over the life of the offering, which has been out since 1998. I have heard the Sun reference implementation downloads are now down to a trickle. JBoss has replaced Sun as the reference implementation. This large market presence stabilizes JBoss, as we get a positive feedback loop from users who send back patches and bug fixes.

The combined impact of JBoss and the two tier-one proprietary vendors is closing the app-server market to all tier-two competition. It used to be that many players could survive in a given commoditized software technology

market by selling to niches. Free Software is changing that equation forever. Today you can customize open source software to fit your niche. The consumer, as a discriminating IT buyer in a down market, is tired of the fifty startups putting out the same marketing, technology and same noise. Simplify, he says. This IT buyer is increasingly standardizing on Open Source.

JBoss is already the de facto standard for J2EE development. We are now leveraging that position to increase our share of the deployment market. Our competitors say that downloads are not the same thing as paying customers. That is true. This is a disadvantage of open source. We do not always know who is using us, nor can we force all our users to purchase our support, as our proprietary competition does. In fact, JBoss is recognized to be of such high quality that people do not always need our support. They often take it as an insurance policy or because it essentially can be used as consulting to iron out their own application issues. A subset of JBoss downloaders that we do track is our documentation buyers. Last year, there were 26,000 "Advanced JBoss" documentation buyers, which if I am correct is more than the total number of customers claimed by the leading proprietary J2EE vendor. JBoss is reaching 100% brand recognition among developers. Even IBM Global Services came to us asking about our partnerships, as many of their customers and developers use JBoss.

So we are becoming a standard, but the question remains: are we, the open source developers, smart enough to establish market standards but not smart

enough to derive a stinking dollar out of it all? Do we have to beg for a living?

Professional Open Source

Well, here is a little known secret about Open Source Java in general and JBoss in particular. The anarchist Sandal Brigade image of Open Source that the media likes to portray is a romantic exaggeration. On average, all of us are clean-cut, highly educated, socially responsible professionals who believe in being financially rewarded for our efforts. In fact, we believe in plain old-fashioned, pre-bubble, profitable business.

The average age of the JBoss developer is above 30, with some graduate students in their 20 and some 50+old timers. Most of us have families and mortgages and tastes that require well-paid jobs. Today, Open Source pays the bills. If it didn't, I'd have to be back to watching my mouth in some corporate bureaucracy. Our word for this is "Professional Open Source." We can work at our passion, be our own bosses and earn a decent living.

We still need to prove that the business model for Professional Open Source is durable. By itself, Open Source technology has staying power. JBoss was around in the heady days of 1999, and VC's wouldn't invest when we could have used it. We survived like cockroaches through the dot-bomb nuclear winter. Now we're bigger, more robust. We are both threatening and repulsive to many proprietary software players. In fact, we have grown wings. In 2003, we are profitable, growing and have a shot at dominating the market.

Profits have enabled JBoss Group, so far, to actively avoid VC money. We don't really need it right now. Those VCs that would not invest when we needed it now want to invest when we don't need it. As opposed to so many fledgling companies in the technology industry that are essentially money chasing a pot of gold at the end of the rainbow, JBoss grew from the bottom up as a real product with real customers. We have a pay-as-you-go model.

Very simply put, all consultants in JBoss, especially those that comprise the "Core," are true professionals, who subsidize their OS development by doing paid services work; they are responsible and moral individuals. At the heart there is still the naïve idealism that inspired us to start working in open source. Our skin is thicker, that's all. And it better be! Professional Open Source gets criticized at two extremes: for some we're idealists who give away our work; for others, we are capitalist exploiters who'll do anything for a dollar. Give us a break. Hopefully we're proving the business staying power of Open Source. As for the people who feel it is somehow immoral to earn a living at open source, they are usually not open source contributors or, when they are, they are usually the salaried ones who don't have to worry about running a business. As for the JBoss Group development team, we are focused on delivering premier technology and making a living by doing this in Open Source. Like moving from a 56K modem to DSL, I cannot imagine going back.

Free software/For-pay service

The secret of how we earn a living is simple, "it's the services, stupid." Our product is free and easily downloaded. Our time and knowledge are not. The middleware field is by nature very services intensive. It is one thing to have a successful open source product; it is quite another for the developers of that product to actually capitalize on that success. I do not see the JBoss model working for Linux or Apache. How much service is there really in desktop Linux? Not much. Is there a ton of service in Apache web servers? Not really. Straight static web servers are not really services, or knowledge intensive. However, once you climb up the stack to the middleware layer, you reach one of the richest consulting fields in the world, supposedly a \$350B/year market.

Carly Fiorina of HP squarely talks about the future of HP being in services. While Ms. Fiorina talks about it, Mr. IBM is busy exploiting it. There is a lot of money in services, enough to support some of the largest behemoths of the IT integration planet, so surely there is more than enough to propel the rapid growth of an aggressively positioned company like JBoss Group. What differentiates JBoss Group is that we are not interested in being all things to all people. We are interested in being the premier app-server consultancy, with JBoss as the premier free application server. Got it? *"Big software companies face the challenge of an inevitable migration towards pricing based on services instead of the software itself"* says John Parkinson, CTO for CGEY US. We do free software and charge for quality service.

JBoss Group's business model is a clean play on services. Our credibility as services providers comes from the fact that we control, maintain and develop a future middleware standard. The core product will always be free. Well frankly, even if we wanted to do it differently, we couldn't. Our license, the Lesser GPL, states that the software will always be licensed for zero dollars, we could charge for distribution (a la RedHat), but we don't, as there is more than enough of a market in services.

Rewarding knowledge

We are looking at something old and something new. What is old is the services business. The wrinkle is that we are a product-related services business, with an infrastructure software product that occupies a key place in enterprise IT architecture. What is new is how we leverage the Net to build a unique, deep and wide expertise in our product, working in a distributed fashion with open source. We give away our core expertise in the JBoss server, thus the real capital we possess is the JBoss brand and the intellectual capital of the expert back office. Intellectual capital, it turns out, is far scarcer than financial capital, even in these post bubble days.

The challenge is to reward that knowledge. The 50-50 development and services model is the first way we reward our developers for their ongoing association with JBoss and JBoss Group. The second way we reward all our developers is through our compensation plan. We reward the open source JBoss development work by distributing economic interest options and annual profit-sharing to the JBoss developers, JBoss Group employees as well as

outside contributors. The JBoss Group employees double-dip as both key employees and regular JBoss developers. It is a simple vision really: to foster loyalty among our development community by making sure all significant contributors are offered a share in the current and future profits of JBoss.

Unique Selling Proposition

The brand and the Net-based distribution by themselves drive business to us. However, the reason we actually close this business is due to what marketing specialists call the Unique Selling Proposition.

It is quite simple. Our services are delivered by the JBoss Expert Back Office, comprising the majority of the developers who write the JBoss software in the first place. What is unique is the ability for clients to be directly in contact with the software developers. By offering the core developers as consultants, support engineers, trainers and documentation writers, we deliver a level of services that is unmatched in the software industry.

Support and maintenance are usually after-thoughts of the modern software assembly lines—staffed by low-paid call center employees. The software industry approaches software the way Ford approaches cars. Bugs are not seen a natural part of software development process, but rather viewed as a failure of the assembly lines. Meanwhile, to be effectively dealt with, the bugs need to be transparent and the software must be accessible—which can only happen through human collaboration in open

source. The vendors don't have the capacity to effectively QA their software, not even Microsoft. While the proprietary software vendors sweep their bugs under the carpet and pretend they don't exist, open source developers know there is no such thing as "Zero defect software."

Bugs exist and are a fact of software. Instead of hiding behind ignorant rhetoric like "self-healing" or "zero-defect," the Open Source community process effectively identifies bugs and enables human collaboration to deal with them. The fact is that for many of our customers BEA failed to remain stable in production, while JBoss passed with flying colors—yet we don't pretend our software is bug-free. IBM, in its corner, sells you "self-healing" pixie dust, the modern IT version of snake oil. Meanwhile, at JBoss Group, we sell honest, high-level human knowledge.

One of the major criticisms of open source software is that it is not supported. In fact, one of the most important innovations of open source is the capacity to grow and stabilize large, complex systems software. JBoss Group offers highest-quality, individually tailored supporting services to its customers. We take support as the starting point of our business model, while others treat it as a secondary milk cow, directly yoked to their licensing agreements. Most of our competition deals with their software bugs and their customers' desire for insurance with mandatory contracts fielded by second-level employees. JBoss Group, in contrast, responds with core developers on an as needed basis. Our service better be good, because it is not mandatory.

Array of services

What exactly are those services on which we make our money? I'll spell it out:

- 1- Training
- 2- Support
- 3- Consulting
- 4- Marketing Partnerships

"Software developers [...] give away for free the very thing they create [...] in the hope that somehow they'll make money selling something else." Says Craig Mundie, VP at Microsoft as a reason why Open Source is doomed. What he fails to see is that another thing we create and is unique is the knowledge of the product. We sell knowledge, and trust me, unlike software that can be easily replicated, knowledge is really hard to propagate.

Training is straightforward. What we offer is unique in the industry: the JBoss core developers take you on a journey through our container design, module by module. This journey will take you "as close to pure engineering as you have ever been" (quote from a Las Vegas training attendee) and you will find it simple. For those JBoss users in the ISV community, the container-level knowledge proves priceless in achieving a tight integration with JBoss. For corporate IT departments, this translates into knowledge that enables you to optimize your JBoss usage. We offer both open-enrollment and onsite training on three topics: Intro to J2EE on JBoss, Advanced Development with JBoss and JBoss Systems Administration for production support. Onsite training can blend modules from the three.

Support is also simple. Due to the stability of our codebase, most of our customers use our support offering as consulting on their own applications that run on JBoss, or in some cases to fund JBoss development for modules they want to prioritize beyond the open source development calendar. As opposed to dealing with a call center, our customers are buying a direct knowledge transfer from the JBoss expert back office to their own developers. Our philosophy is that companies should invest in knowledge rather than in empty licenses. We offer both remote and onsite support and consulting.

The JBoss Software Marketing Partnership program is targeted at software vendors whose offerings run on top of, or are complementary with JBoss. We enable them to leverage our brand and 2 million downloads to reinforce or establish theirs. Even established partners, like abaXX, a popular European portal vendor, reported a 700% increase in website traffic once they entered into a partnership with us. JBoss enjoys tremendous prestige in the development community as the de facto enterprise development standard. This program allows our software partners to raise their profile with an influential crowd of developers.

Conclusion

I remember a “Business of Open Source” presentation I went to back in 1998 at Stanford. The presenter, some editor from “Dr. Dobbs,” said that one of the problems with the business of Open Source is that it will remain a vague proposition for mainstream America until there is an Open Source billionaire.

If JBoss Group is successful; it will mark the end of a billion dollar licensing business in application servers, a business represented by the likes of BEA Systems and its CEO, Alfred Chuang. In a recent interview, Chuang tried to dismiss JBoss as being irrelevant. The fact that he and others at BEA spend so much time trying to convince people they’re better than JBoss belies the fact they are running scared. No proprietary vendor can compete with JBoss’ TCO (total cost of ownership).

TCO = hardware + license cost + services. *“The majority of the TCO comes from services [...] so free often turns out to be more expensive than it seems”* says Scott Dietzen, CTO of BEA. We agree with the first part, disagree with the second. Hardware is becoming commoditized, but should be factored out when comparing app-server vendors. JBoss license cost is zero, even for massively clustered systems. Service is pay-as-you-need and of higher quality than that offered by our competition. The best argument Chuang made was when he claimed he was selling “peace of mind.” What an argument. You’re not buying technology, you’re not buying a product; you’re buying “peace of mind” at \$10k per CPU. They have stopped arguing technology quality, features and TCO. Our competitors can’t win these points and they know it. I also read that Mr.Chuang personally owns 12 Ferraris and gives them out to his top sales people.

Whether the JBoss open source business model sustains Ferraris or VWs for its sales people remains to be seen, but one thing is clear. I must be incredibly crazy not to be in the “peace of mind”

business, right? I will be soon. "Open Source software puts at risk the continued vitality of the independent software sector" says Craig Mundie, VP at Microsoft. We disagree it is shift. Professional Open Source offers a business model where the successful contributors to the project have the option of joining the business and eventually earning a living comparable to that of partners at large law, consulting and accounting firms. And that's good enough for me.

So no, I don't have a vision of Open Source as wish fulfillment, some magic ground that you can just "throw" a project into and, overnight, it will sprout into a tremendous success. The qualities that have made JBoss Group an open source success are the same ones that characterize successful non open-source companies: hard work, perseverance, organization and financially rewarding your contributors. I don't have any promises of quick, easy money to offer prospective open source developers, but what I can offer is a clear-eyed and realistic vision of living well doing what you love. To be your own boss, to work with your peers, to freely invent the technological future and to earn a decent living while doing so, this is all worth a lot. But in the words of MasterCard ad campaign: taking on an industry with 30 individuals. Well, that's just priceless.

To be sure, the tectonic shift Open Source and the Net caused by enabling cheap development and free distribution in the app-server industry would have taken place regardless of the players. The accomplishment of JBoss and JBoss Group is to harness that energy and build a viable business around it.

This is why it is ridiculous when people accused of us of wiping out millions of dollars in market capitalization overnight for one of our competitors. I assure you, if I had that power, I'd be making millions in the stock market, not giving away my core competency for free in Open Source. We live in the post-Dot Bomb era. If those dollars vanish and re-appear overnight, one wonders how real they are. What is not up for debate is that software purchasers can no longer afford to ignore value. The proponents of the software licensing model justify for-pay licensing as the necessary pay-off for large investments in research and development. Where that expenditure is necessary and justified, I tend to agree with them. However, where quality open source alternatives exist, proprietary software will not be able to compete. I don't deny that we are witnessing the end of the billion-dollar application server licensing market. Nevertheless, it is also equally true that JBoss, as a free, high-quality application server, is allowing companies around the world to save that billion dollars in their IT budgets. That money then becomes available to be invested in their own people as well as in the higher tiers of software that are not affected by Open Source. "*A world of applications built on Open Source programs will finally thrive*" says Alex Salkever of Business Week Online in March 2003.

I am a big fan of Sim City, and not just because Maxis Entertainment runs Sims Online on JBoss. What you learn in that game, and in the urban planning statistics on which it is modeled, is that cheap, high-quality infrastructure is necessary to build the technology "cities" of the future. Right now, we're just living in caves.

The perceived “safe choices” of today will not be the “safe choices” of tomorrow. I have seen the future and it works. This is the future where people will say “No one gets fired for choosing Open Source.” This is a future where JBoss Group will also get its fair share of the “peace of mind market.”

Soon, one day, we will all walk through walls. Not just me, everybody.

marcf,



Dedicated to my father, Daniel Fleury, author of the JBoss Group Compensation plan and without whose involvement none of this would be.

Note: This paper does not claim to be an exhaustive treatment of open source business models, the business models are many and we are all doing microeconomic digging establishing their shapes. JBoss Group happens to sell services in a technology niche that is services intensive. Other companies attempt to leverage Open Source with strategies like dual-licensing, subscription, packaging, or proprietary software on top of Open Source. If they are profitable on a different model, more power to them. Those companies are capable of telling their stories; this is ours.