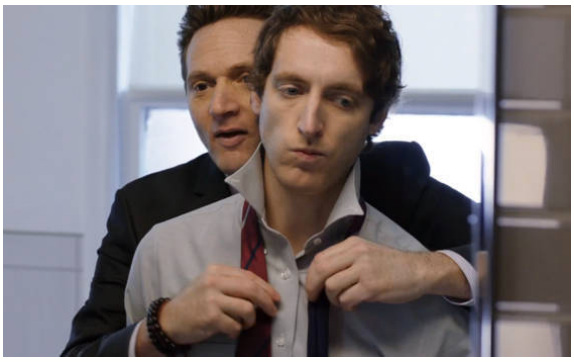
The Register®
Biting the hand that feeds IT

Ever seen a storage startup and thought 'Pshaw. I could do that?'



You read The Register about people getting ahead with the most ridiculous startup ideas and ask: "Why that can't be you?" How is the game played? Well, let's design a storage startup from scratch and find out.

I'm picking storage as the market for my hypothetical start-up in part because I have worked closely with a number of storage startups over the past few years. The other reason is selfish: I honestly do have some good ideas for startups and I don't want to give away what might be my own personal golden ticket.

The storage market is largely played out, so the discussions about the tech are pretty high level. This probably isn't going to help you actually rush out and get a startup off the ground, so please do not take this as a blueprint. It is intended only as a look at what the journey of a startup is like.

Assembling your team

To start off with you'll need about $100,000 USD and you'll need to know some people. The $100,000 is to cover the living expenses of the core team of three people for three months. $100k/yr is $8.3k/month. Three people @ $8.3k for three months is ~$75k. You'll need the other $25k for plane tickets and schmoozing.

Anyone worth having on the ground floor is either going to be in a position that they need at least $100k/yr to pay expenses, or they've cashed out already and simply don't need money. Don't skimp this. If you can't find a way to pay the bills of the core team until the Angel round, stop right there. The whole endeavour will simply collapse in acrimony and recriminations before you get anywhere.

Most folks don't typically have a spare $100k around, nor the ability to bootstrap for three months, so this usually means the startup consists of four people, with the fourth being the pre-angel funder. In most cases this individual doesn't concern themselves directly with running the startup.

There is certainly room for disagreement. In penning this piece, I reached out to a number of the tech industry's best and brightest. Many responded, a few offering worthy advice, though several requested anonymity for various reasons. Where and why these luminaries disagree with my take is certainly worth consideration.

The first of our anonymous executives, whom I am unimaginatively naming John Smith, is director of marketing at one of the largest tech companies in the world, and a proper storage geek in his own right. Smith offers this thought: "I am not convinced about your 100k argument. The developer and glad-hander are typically veterans, and together are True Believers in the technology. They are on a mission to show that they can change the world and get rich. As such, I think that they are most focused on stock ownership because they think that they will be crazy successful. Thus, while they would like to get paid, I am not convinced that it is as critical as you imply. (Remember that this is only during the initial three months.)"

The truth is that how much money you need to start is going to depend on whether or not your startup team has bills to pay, and how much they are willing to sacrifice to make it happen. Equity is an important consideration and is critical to making startups work. Who is willing to work 100 hours a week for terrible pay? People with stock options.

Written by Trevor Pott - trevor.p@egeek.ca

Assembling a crack team of startup heroes

I am assuming those who are reading this don't have the option of simply not worrying about money, so we'll stick with the idea that you need to have some pre-angel funding in order for your startup team to survive.

Do not think you are going to create a startup "in your spare time" while working another job. That's so statistically unlikely as to not be worth discussion. Not to mention a lot of companies "own" any ideas you come up with "in your spare time" via fairly evil contracts.

So we have an idea of the startup's initial requirements, but why do we need all these people?

Roles and responsibilities

If there is disagreement about how much money you need to assemble before you launch, things can get really contentious when we talk about the initial composition of your startup team. I've watched a lot of startups launch and fail. I've seen a few succeed. I've seen some get bought out – but for fractions of what was wanted – and I've seen startups torn apart by management changes.

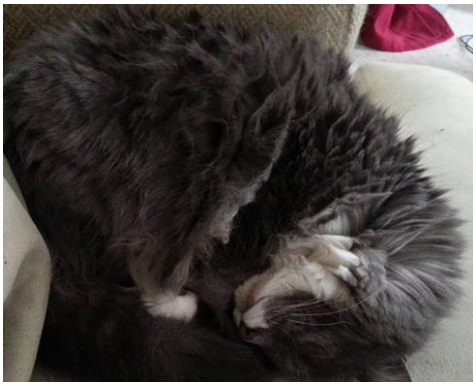
Based on my own analysis, the three people you need to start with are **a technical writer, a kickass developer and a glad-hander.**

The easiest role to understand is that of Kickass Developer. Whatever it is you're building, they're the one that is going to have to make it happen. All of the core members of the team will have to have input into the design of the product, but converting concepts to code is going to take someone special. The kind of special that can code for 16 hours straight, 6 days a week for three months without going mad.

No angel investor is going to invest until you have something worth investing in. This means you need a prototype, no matter how crude. As we're designing a storage startup, let's look at what that means.

Written by Trevor Pott - trevor.p@egeek.ca

Most storage today is pretty simple to use. You get a nice GUI, there's an API and there's probably a command line. The GUI is web based and probably does everything through the API. The command line is probably also just a means to call the API, so that API is at the core of your product.



Management hard at work

Kickass Developer needs to build a prototype version of your storage dream in three months that both demonstrates the basic viability and the fact that it can be configured and manipulated via API. Any decent angel investor will by now understand that once you have that API, the window dressing can be farmed out to comparatively low-cost devs.

But you need that core product working. Some features can be added later but the core needs to be there. In the case of our storage startup, data services and the like are good choices to leave to the angel round, but we'll need a good object store and some means to provide for data resiliency, a presentation layer (so things can consume the storage), and a Unique Selling Proposition (USP) that gets investors to fund you instead of the next individual.

Gladhander exists to schmooze. Someone has to sell the idea, the people, the technology and the USP to the investors. Eventually, they'll also be responsible for schmoozing keystone customers. A successful startup will have Gladhander act as a firewall between the money men and everyone else.

Gladhander will also be the individual running interference with the accountants, the lawyers and everyone else. This is part of Gladhander's second real job: ensuring that Kickass Developer and technical writer have the lowest possible level of anxiety. Productivity is the most important thing at this stage, and nerds aren't particularly productive when they're anxious.

Technical Writer's job is probably the most nebulous. At this early stage of the game they have three jobs, all of them vital. The first is to understand what it is Kickass Developer is coding, how, and why. The second job is to document what they're doing so that the company doesn't run into "success failure."

Technical Writer is to product copious amounts of technical marketing for Gladhander: whitepapers, blogs, PowerPoint slides, case studies and business plans. The technical writer is the translation matrix between Kickass Developer and humans.

The natural roles for these individuals, once the funding is in, are to have Gladhander be CEO, Kickass Developer be CTO and Technical Writer be product manager. Technical Writer will have to serve as the head of marketing (with whatever title you prefer) until the A round provides enough funding to get a real marketing lead.

The subject of who should comprise your core team is a hotly debated one. It's worth pointing out that I'm just a Canadian nerd and I haven't gotten rich doing any startups. Other people who have actually played the game (and won!) disagree and their views should be considered.

Jeff Ready, CEO of Scale Computing, offers the following thoughts: "On your original startup team of three, there is no sales person. Perhaps this is implied in your Gladhander, although you already have them doing quite a bit. I would take a salesperson over a tech marketer in that initial group all day long. One, because I can write the marketing stuff myself if I need to, but more importantly because customers are the key to the future funding."

Written by Trevor Pott - trevor.p@egeek.ca

Only the best of desks for the CEO

John Smith, (pseudonym I'm using in this series for the marketing director of the largest tech companies in the world) offers similar advice: "The biggest one is that I am not convinced of the criticality of the tech writer in the formative stages. In my experience, the key is the Ktickass Developer and the Gladhander, the latter of whom serves as the tech writer early on. In order to keep things skinny, the Gladhander typically does the schmoozing and creates the messaging and similar related content for the investors while the Kickass Developer focuses on the plumbing."

Here I feel I must poke my head in to these excellent comments. They aren't wrong: sales is absolutely critical. In fact, I view this as a key function of Gladhander: schmoozing those as need to be schmoozed and gladhanding the high rollers.

I think a CEO who can talk Venture Capitalist, sell things and crack the whip over the employees is a much easier-to-find combination than a CEO who can talk Venture Capitalist and write decently. I've encountered plenty of CEOs who could sell snow to an Inuit. On the other hand, I literally wouldn't have a roof over my head if most CEOs (or marketing VPs) were intelligible at the keyboard. Converting CxO into human is my day job.

An anonymous senior vice president at a one storage startup, who I will call John Johnson, had this to add: "There are almost always at least a couple developers. The other key to the early core of the company is you need people with cross-functional skills. Product manager / developers / content creation (slide decks are critical) / pitch person / business development (relationship person with the rolodex) / channel expert (most startups these days work with channel partners) skills all need to be included in that core team."

Ugly babies

Do not let Kickass Developer be CEO. Do not do. By this I am not advocating that CEOs can't be engineers or developers, but rather that the CEO can't be the lead developer of the product this startup is working on.

The problem is that a huge part of the early life of a startup is seeking and incorporating feedback. Many experts I reached out to pointed out that they would seek out potential early customers before ever considering putting together money for a startup and this must continue for years if success is to be achieved.

This process of engaging with customers will lead to changes in the design of the product. If the product is the CEO's baby, there's a problem. Nobody likes being told their baby is ugly and when the person in charge is emotionally invested in a specific solution, startups usually fail.

Again, there are exceptions. Dheeraj Pandey, CEO of Nutanix explains, "I couldn't relate to the Gladhander. As a CEO, I wrote a ton of code. Every partnership is different, you know. Just like a family – parents, siblings, etc, and making money is the last reason why people build sustainable companies. The biggest reason relates to the "right brain" – autonomy and an elusive desire to change the world."

As one storage VP puts it: "the delicate Balancing Act of early startup life. Too much passion and there is an argument and a critical team member leaves. No enough passion and the plane flies into the side of the mountain. Too much debate and the product never ships. Not enough debate and the plane flies into the side of the mountain."

Written by Trevor Pott - trevor.p@egeek.ca

With the who and how of starting our startup sorted, let's take a break from people and politics and dive into what the potential technology of our startup might look like.

Tech 1.0

Before any of the funding or shenanigans above, you need an idea for a product. As we're designing a storage startup, I'll lay out a storage product. To design this product, I'm mooshing together ideas from dozens of storage startups alongside my own, so I'm sure lots of folks will see something of their own startup reflected in this design.

In a perfect world we want to build a storage solution that meets all needs for all people. Everyone else will simply stop using the competition because ours is so awesome. Boiling the ocean, however, never works as a starting proposition, so we'll focus on one primary problem.

NVMe is all the rage, and faster is typically better, so the core goal will be to build a storage product that goes really fast and takes advantage of what NVMe has to offer. NVMe is good for fast storage primarily because of high queue depth.

Let's start with the new Intel Broadwell-based Xeon-Ds with integrated FPGA. Customize the FPGA for your high I/O workloads. This can help get around the part where NVMe is really, really CPU intensive. Get a NIC that can handle offloading without imploding in order to also drive CPU usage down. Pack it all into one of Supermicro's 48 drive NVMe chassis and slap your brand on it.

As part of your sexy software you build a scale-out storage solution similar to Coho or Solidfire so that you can handle not only failure of the device, but failure of (at least) a node. Let's say it's object based and allows for n copies of an object.

Qualify a good switch as part of the solution, one that can handle multicast really well. Use multicast for transmitting reconvergence data and other inter-node communications. This makes rebuilding of nodes easy and low-impact, even with huge NVMe setups. The rebuild stress ends up shared equally across the cluster, but the whole of the cluster doesn't have to play silly buggers trying to figure out who is doing what. The good switch is necessary because cheap switches are pants at multicast, and you're going to need every scrap of networking you can eke out.

Build locality awareness into the storage platform and make sure any node can serve up storage. If you have a 32-node cluster, use 16 virtual IPs so that when you assign storage to a requestor they can not only manage to load balance requests across the whole of the cluster, but the cluster can make sure that the assigned storage has replicas physically on the nodes responding to the storage request

Most of this can be done by welding together various open source projects. As a 1.0 to prove viability to an angel investor that very well might be good enough, assuming you can demonstrate that you have identified pain points and have a solid roadmap for replacing some (or all) components by A round. You need something to patent, and the investor doesn't want to pay you to make a start-up anyone can come along and clone.

Be wary of other startups! As I discussed above, this design - like every storage startup – is taking elements from various existing companies' products*. Make sure you know who has patented what, and you work around that. Get Gladhander to earn their keep!

Written by Trevor Pott - trevor.p@egeek.ca

Knowing what you want to build, however, is not enough. You need to be able to build it in stages, monetize it at different points, and have some idea of how you will handle investors, staff, partners, community and politics.

How to make it through the crucial angel-investment round

The angel round is key. In today's world of tech startups, it's not unheard of to get $1m in angel funding with the expectation that this will last you about a year. The goal of the angel round is not to produce a sellable product. The goal of the angel round is to produce an investable product. These are different things.

A sellable product is more than just code that works on devices that don't fail. A sellable product has an entire support infrastructure behind it. It has a sales channel functional enough that customers buying the product can be assured the company is likely to still be around until at least the end of the current refresh cycle.

Startups burn investor money to survive. That's a fact of life. But a startup with a sellable product is not 100 per cent reliant on that investor money. They have cashflow from keystone customers and an organized sales and marketing apparatus bringing more in all the time.

An investable product needs to work, do so reliably, and there should ideally be around five keystone customers using the product. These customers won't likely have paid for the product. Most will be using the product in test and dev only, but before you hit up the angel investors try to find at least one who is willing to put your product into production, even if it is in one branch office that nobody cares about.

With ~$1m to burn in a year Kickass Developer will be getting some friends. Expect the development team to be four devs, in addition to Kickass Developer, and one very overworked Documentation Monkey, who documents the code.

Documentation Monkey is needed so that Kickass Developer doesn't have to waste months onboarding every new junior developer. Code is documented, styles, policies and procedures are documented and new developers can be added into the mix without hurting deadlines.

Technical Writer will take Documentation Monkey's work and build a formal API manual for both internal and external use. Technical Writer will also keep cranking out newer and better salesy and marketing work for Gladhander.

Gladhander will be spending most of his time briefing tame analysts and journos, meeting with potential A round investors and cracking the whip over everyone to get things done. Their job is also to work with both Technical Writer and Documentation Monkey to find something – anything – they can patent. The more patents the better. Patent submissions probably won't be granted in time for A round funding, but they help a lot in securing a solid A round.

Johnson has worthwhile insight: "The typical time from inception to General Availability (GA) is 18-24 months. This is how long the VC community will fund a startup before they want revenue. The company will not be close to profitable, but they need to show the product works and customers will buy it to get the next cash injection. This 18-24-month window creates a limit of the scale of the problem the typical startup can address."

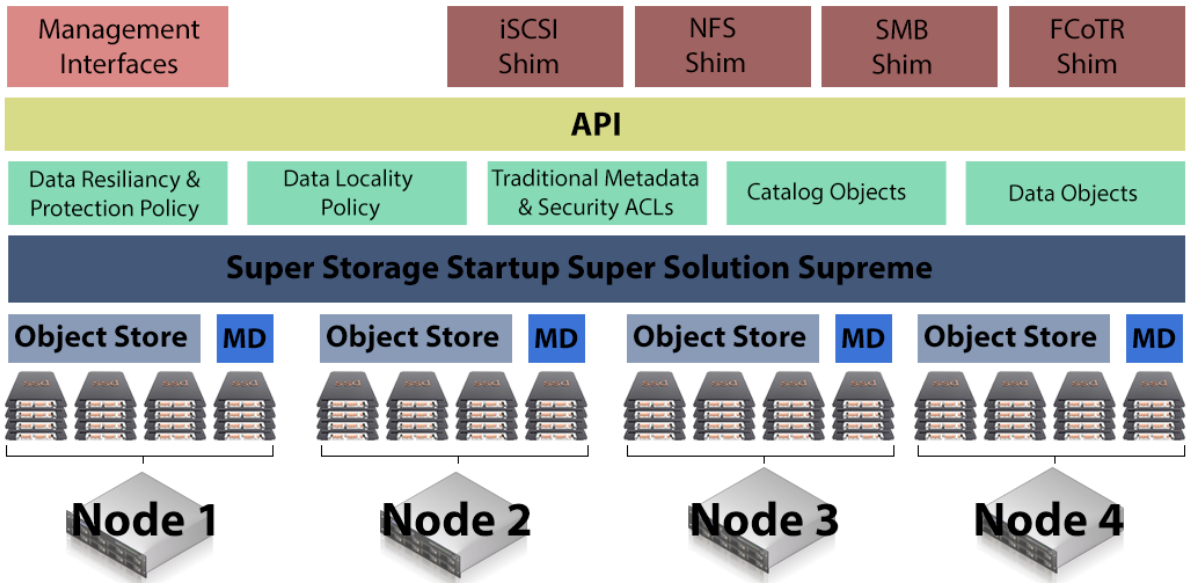
Written by Trevor Pott - trevor.p@egeek.ca

Tech 2.0

Tech 2.0 is all about the evolution from that welded together open source solution towards an investable product. For this you'll need a management UI that makes people say "wow". Kickass Developer will be noodling about under the hood replacing or extending the open source packages, while the minion devs will build a UI on top of the API.

The two core pieces of intellectual property our storage startup will develop here are going to be the code that accelerates storage using the FPGA in the Xeon-D and the metadata system overlaying the object storage.

The FPGA code allows for faster/stronger/smarter/better nodes that serve more IOPS while consuming less power. This is important, because while everyone else is lowering costs with hyeprconvergence, our startup is going to try to convince customers to invest in storage separate from compute. Our only means to lower costs is to offload as much of the storage calculation as we can to the FPGA in order to cram ever more NVMe drives into a single chassis.
The metadata is almost meaningless at the early stages, but is the key to everything in the long run. We'll get our keystone customers by virtue of nothing more than going really, really fast in a very small foot-print, but that advantage won't last.



Generic storage startup gets generic storage diagram

Building a flexible, extensible metadata engine right at the outset allows us to do neat things later on. As the 2.0 is all about the sprint from angel round to A round, the metadata we store is going to be pretty similar to what most file systems store: object size, name, ACLs and so forth. That will change.

Once a management UI exists that isn't worthless and the underlying storage product does the job of going really, really fast while still being reliable it's time to get those keystone customers and chase the A round!

The A round

You have a product that works. You have a product actual people in actual companies can and do use. Yes, you had to give the product away for free and bribe the people involved to get it into production, but that's all in the past. You've got an A round, probably about $10m, and the faith of investors that you can take this all the way.

Written by Trevor Pott - trevor.p@egeek.ca

Here you can expect management changes. Livens explains: “The other thing that I ponder is the longer term role of the Gladhander. I think that typically an early stage GH has critical skills that are vital to success, but a different skill set is needed as the business scales. They are a visionary with the drive and enthusiasm to spell out a vision and get others (internal and external) to follow. However these people are not typically operational in nature. There is an element of beancounting that is required as CEO that these folks are typically not skilled at that. Of course, that is where a strong COO could come in and so maybe they become CEO, but long-term, I see them moving to other roles.”

Our storage startup now needs to build out the sales and marketing apparatus. While the nerds beaver away on the 3.0 product that will lead into the B round, the current version has proven good enough for some use cases and so it’s time to hit the convention circuit!

Get customer buy in! (That picture takes me back...)

An A round startup isn’t hitting the convention circuit desperately seeking leads. Its still early days and 10 of the right customers are better than hundreds of the wrong customer. The startup needs customers that can push the technology to its limits, are personable and friendly (so they won’t sue every time they find a bug) and who will ultimately bring in more revenue than the cost of support.

More than anything, our storage startup has to find customer advocates. These are customers willing to talk about how great the storage solution is to investors and potential customers, even if those discussions need to be NDAed.

The A round is about building a portfolio of customers, analysts and tech press who fawn over you and convince everyone else you’re the greatest thing since sliced bread. Sales is important here, but marketing and PR win the day.

The technical marketing manager told me start ups build what he called a “major crescendo of community buzz” up to their next major release. That consists of customer advocates, publicly facing corporate cheerleaders (aka “the evangelist”), and community influencers who have gotten a good look under the covers and get really excited for the future possibilities of the product.

This has what he called a huge multiplier effect for startups.

- Its awareness is critical for a new vendor in a crowded ecosystem
- It provides a rolling start for the demand gen and influencer marketing engines
- Excites early stage prospects to want to be a part of this very exciting party,
- Indicates to the market (and all the investors watching the market) that ‘we got something good going here’

Getting the name out there, getting viewed as a disrupter/innovator/all around cool company, and having the community falling in love with you early on in the product cycle is never a bad thing.”

In the background, the nerds are working on the features that will drive the long term viability of the product. Kickass Developer is doing less and less hands-on coding and more architectural level work. Gladhander hasn’t slept in six months and Tech Writer has gone back to the beginning on all their content because Tech 3.0 is going to change everything.

Written by Trevor Pott - trevor.p@egeek.ca

Johnson has a warning for all: “Shipping the product is just the beginning. Everything is 10x harder once the product ships. Support needs to come online. There will be bugs. There are always bugs. Support will be new, so engineers will be involved in the escalation process. The more time engineers spend helping with support, the less time they are coding. It is a whole new world.”

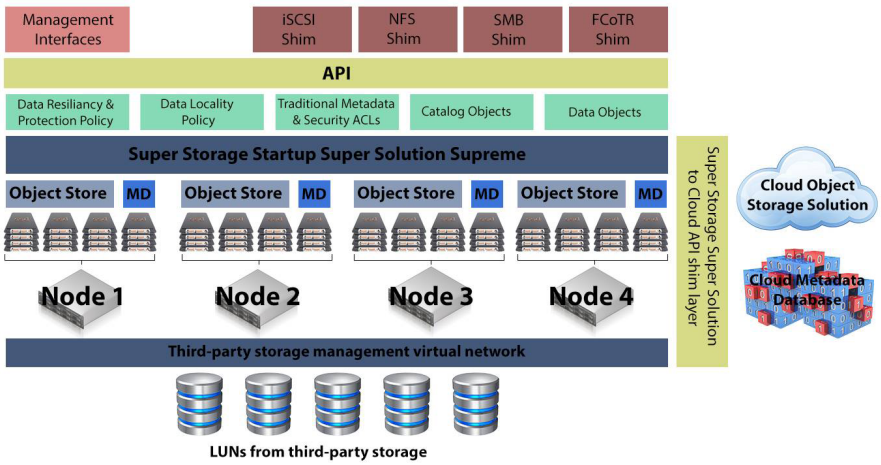
Tech 3.0

With Tech 3.0 our storage startup is going to stretch to the cloud and embrace third-party storage. The core product is still super-duper deluxe NVMe storage of fastness, but the ability to put data in the cloud or on third party storage opens entire worlds of possibility.

Not everything needs to be on our NVMe rocketship. The ability to place cold data on cheaper storage or archive off to the cloud means that instead of a niche player addressing the needs only of speed freaks, our storage startup can now get into brownfield environments.

We get into these existing installations not by killing them, but by co-opting them. Allow customers to subject their existing storage to the NVMe cluster’s clearly superior management interface and make their lives easier. You still need to buy one of our crazy NVMe clusters, but you don’t have to throw away all your old storage when you do so.

Bear-hugging third-party storage also allows our storage startup to embrace new data protection schemes. Now you can copy data to other storage units or even up to the cloud. Infrequently used data can be tiered, data resiliency that requires data on device from multiple manufacturers can be built in and more.
Take the data locality work – done back in Tech 1.0 in order to solve speed problems – and extend it using the metadata engine we designed, but never really explained the use for. Now our storage startup has a product that allows customers to “tag” objects with locality information.



Generic storage startup diagram, now with more cloud!

Objects can be individual workloads (LUNs), or in the case of NFS and CIFS shims, individual files. Now we can do neat things like tag “this file/workload cannot be stored in an American data centre”, etc. This is essentially done by extending the metadata engine used to store the objects on the cluster, and opening an API so that DevOps teams, service providers and so forth can flag files in an automated fashion.

Tech 3.0 just became something that service providers absolutely must have, banks will fall all over themselves to get hold of, and CDNs will start paying attention to.

You need customers, but credibility too. One begets the other. But which comes first? We are now in the final, challenging chapter of early start up mode: of getting paying customers and leveraging them to your advantage in the world to gain more.

Written by Trevor Pott - trevor.p@egeek.ca

As Tech 3.0 launches the dance for the B round starts. Everything here depends on how much interest can be drummed up. Interest is going to stem from two things: credibility and awareness.

Credibility will largely flow from the keystone customers you have managed to wrangle thus far. It can stem from executive credentials such as academic degrees, work history or who vouches for whom. It can also stem from having solid roadmaps. This feeds into awareness. Awareness is all about understanding what the market wants, what you can reasonably deliver and who is going to try to kill you along the way.

For our storage startup, an ecosystem will appear whereby third party applications will scan the contents of files for things like credit card information and mark locality information into the extended metadata using the API. The super NVMe speed awesome might be the core of our storage solution, but what's going to sell it to the masses is that metadata engine.

Even rivals can be friends

Tech 3.0 brings with it the ability to flag individual objects/LUNs/files not only with data locality information, but with QoS information. This in turn could affect locality if QoS is provided by chucking low tier stuff onto a rust array, but that's an implementation detail. At some point the metadata engine would be evolved to the point of allowing per object data resiliency information to be encoded.

This is where the third party ecosystem starts to explode. Big data eggheads, security nerds and everyone else will want to get in there, scanning files/objects/LUNs to see what matches their criteria. Then they will want to flag those objects with "store here, make it this fast, make it this redundant, back it up this many times, to these mediums", etc.

At this point we've created the ultimate file storage system – with third party data nerd help – that meets corporate needs, but because it's all controlled by robots no human being could ever hope to keep in mind what files are physically where.

Politics

Now customers are locked into our storage startup's platform. The key is to make sure that they don't really feel that way. If we were smart, then when we embraced the cloud we did not do so by storing files there using an overlay. We would have addressed cloud storage using an API, thus storing the objects natively. Similarly, any third-party storage that has a true API for storage is storage we'd talk to natively instead of using an overlay.

In this fashion we can point to the fact that a small change in parameters and our storage system would actually self-evacuate all the storage onto other systems, where they would be accessible, and thus we aren't locking anyone in. Our storage startup not only isn't locking you in, it provides all the mechanisms to transition seamlessly to other people's storage – if that's what they want.

Of course, Behind the Scenes, everyone knows that's rubbish. Our storage startup's ability to use those Xeons with FPGAs and provide stupidly ridiculous NVMe storage is really the cornerstone to why everyone chose us in the first place. For all that old and slow data lives elsewhere, they're addicted to our NVMe rocketship and they're unlikely to replace that.

Deflecting lock-in concerns should be easy. Our storage startup is two or three layers beneath the actual lock-in, which is the third-party software that scans everything and meets business needs by addressing our API.

Written by Trevor Pott - trevor.p@egeek.ca

In the background, we're quietly working on Tech 4.0 which includes our own big data scanner. This allows us to "meet customer demand" for removing the "lock in" of the third party scanner. (A concept of lock-in we created with subtle marketing and that, by now, the company behind the third-party scanner has figured out they can sell for a metric yoo-hoo because all these customers are captive).

Tech 4.0 rides to the rescue by freeing customers from the third-party pain point we created. We anger our ecosystem partner(s), but who cares? All they did was talk at our API. It's our storage startup who actually stored the metadata and the objects. The third party company was always disposable.

For a short while, we're the good guys! We've eliminated the pain point of the third party data scanner, which we can rightly claim was created out of necessity. We're a startup and didn't have the engineering resources to boil the ocean back at Tech 3.0.

In removing that pain point we make Friends For Life amongst the now massive user and enthusiast community that surrounds our product. Our conferences (we're probably at our third or forth by now) are a love-fest.

A couple of years later customers will figure out that we now have an enormous vice around their balls, but it doesn't matter anymore. Our storage startup's valuation is somewhere north of $50bn and the IPO is closing in.

Parting advice

Johnson has some food for thought. "Startups are a full contact sport. There will be debates and arguments. There may be yelling and people may quit in the process. The early days are really tough, but those lines of code and product decisions lay the foundation for the future of the company."

Ready offers experience in dealing with investors: "Watch a typical political speech. 'Ladies and gentlemen we have a problem with zombies, they are everywhere, they cause problems and it's terrible. I, the great politician, have a solution to this problem. if we provide funding for the zombie toast-a-matic, it will eradicate this blight on society!' At this point everyone is dozing off, as are your potential investors."

He continues, "Let's change the conversation a little. 'Now, Mrs. Betty Sue Jones had a problem with zombies. They rose from the ground and killed her cat. Betty – will you please stand up?' camera pans to Betty in the audience 'There she is folks, someone just like you, living in your neighborhood. And her cat is dead because of these zombies!' catch tear rolling down cheek of Betty or another attendee 'Thank you Betty. Now folks, how can we not do something here. There are 20 million Betty's just like you and me, and we all have cats. The time to act is now.'"

Ready says "Right about now the entire audience is saying oh my god take my money don't let my cat die! It's the anecdote that connects the logic (or illogic in many cases) of the problem/solution statement with the audience/investor. And yes, the investor then is thinking of investing based on one or two anecdotes more than the 'market size' — but that's how it works."

Ready rounds out his advice with "To be clear, I'm not implying the investor ignores market size because of the anecdotes. Rather I'm saying that the investor is taking the one or two anecdotes and extrapolating that into an overall market size. They will look at two anecdotes and use that to decide your solution could fill the market. It's the anecdote and emotional connection that are what actually sells, not that dry and boring solution + market size equation stuff. Sell the sizzle not the steak, so to speak."

Written by Trevor Pott - trevor.p@egeek.ca

A third anonymous storage VP, "just in case this triggers a round of stump-the-chump or pop quiz at a local watering hole" offers up some book suggestions to those interested in making a serious go at their own tech startup:

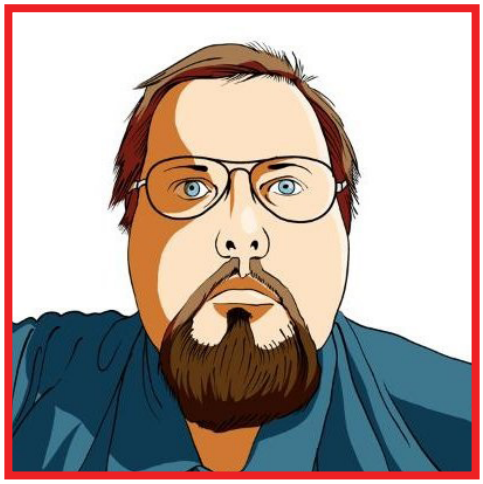
- Steven Stralser – MBA in a Day
- James Brannock – Business Case Analysis
- Jim Collins – Good to Great, Built to Last, Great by Choice, etc...
- Donald Mitchell – The Ultimate Competitive Advantage
- Gary Donahue – Arista Warrior
- Dave Hitz – How to Castrate a Bull
- Tony Hsieh – Delivering Happiness

The happy ending

We've played the long game, created a series of viable incremental product deliveries that generated revenue, gave a clear path for advancement, had a roadmap, were aware of the likely political and community pressures and this demonstrates to Wall Street that they should shower us in gold. Achievement unlocked, we cash out and wait two years before stepping down and buying our own Mars colony.

And that, dear reader, *is how you get rich in Silicon Valley*.

About the Author

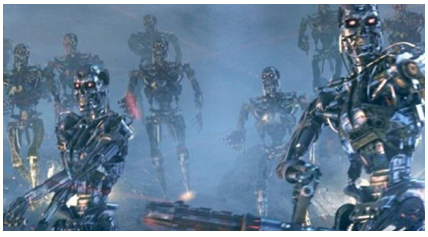


I am a both a systems administrator and a technology writer. I have been fixing computers, networks and associated bits that go "ping" for a little over twenty years.

As a systems administrator, I have focused primarily on small business and the midmarket. This is the kind of tech work that doesn't allow for specialization. It forces sysadmins to be generalists, and it allows us to see a little bit of everything in use at some point. As a writer, I have spent much of the past five years investigating, testing and writing about every kind of storage you can imagine.

Thank you for reading, and I hope you enjoyed it.

More from Trevor Pott



I, for one, welcome the rise of the Infrastructure Endgame Machines

Setting an ioSafe 214 on FIRE

The Register guide to software-defined infrastructure

Written by Trevor Pott - trevor.p@egeek.ca