Previous post

With Sync Solved, Dropbox Squares Off With Apple's iCloud

By Rachel Swaby
Email Author
December 22, 2011 |
6:30 am |
Categories: Entrepreneurs, Guest Column, The Cloud

In early 2009, just months after Drew Houston and Arash Ferdowsi launched Dropbox, a service that promised to sync users' photos, music, and documents across computers, Apple's MobileMe team suggested the pair come by for a chat. The two MIT-schooled twenty-somethings knew they were in for some discussion about how their little startup was competing with the computing giant, but they were not prepared for what was first off the block:

How did you get in there?

The question was about Dropbox's integration into the Finder — something Houston and Ferdowsi accomplished by hacking their way in. In order to get highlighted text to accompany a file that had been updated within Dropbox, the founders had to break into the Finder's processing server and insert their own code.

But despite the breach, the Apple inquiry wasn't an aggressive interrogation. It was actually an honest inquiry. Even though Apple's Finder team was working under the same roof for the same company, MobileMe was refused access. Houston's response came with a smile: "Yeah, it's kind of hard to do."

"We had to solve a lot of pretty tough technical problems," recounts Ferdowsi. Houston is a little more emphatic: "We had to reverse engineer without source code — basically open heart surgery on the Finder to discover the assembly language routine that draws icons — and then squeeze in our little icon. You

have to do similar but different things on Tiger, on Leopard, on Snow Leopard, on Lion, thirty-two bit, sixty-four bit — every presentation. It is exhausting."

From the very beginning, the founders of Dropbox knew they needed to tackle a set of massively complicated problems — problems that had flummoxed computer scientists for four decades and complicated the lives of computer users for as long as they'd had access to more than one workstation. Getting a set of files on one computer to match the files on another seems, on the surface, to be a simple enough task. But actually accomplishing it means scaling an established list of academic thought problems and rabbit hole engineering challenges.

For starters, how do you get files to look and act the same on platforms that don't want to work together? Then, how do you make sure all copies of the same file are identical if multiple people are working on it simultaneously? What if there's no internet? How do you get access to files in the cloud when you're not connected to it? And even once you've scaled those significant hurdles, how do you tuck away all that hard work and make the experience nearly invisible for the user?

In just a year and a half of late-night coding in crappy apartments and small, smelly offices, Houston and Ferdowsi had checked off the whole list. By collecting and tweaking existing solutions to a disparate range of computing problems, they created a package that could be bundled into a clean and lean release. The result is a service that has gained 45 million users since its September 2008 launch, providing a solution for people that didn't even realize there was an alternative to packing thumb drives and e-mailing files to oneself.

But Dropbox's success is now at a critical moment. Providing a way to give users access to everything they do on a computer from everywhere has become the brass ring for digital services. And Dropbox is the one to beat. After years of remaining largely below the radar of the world's biggest computing companies, Dropbox is now faced with a host of very prominent competitors. The most immediate, of course, is Apple, whose iCloud service launched in October.

iCloud hits Dropbox hard on the simplicity and ease of use fronts because Apple has been working in this space from the beginning. iCloud may be cross platform averse, but its automatic syncing slices seconds from Dropbox's drag-and-drop approach. And then there's mounting competition from a whole slew of young companies also willing to give up sleep and a social life for a piece of Dropbox's $4 billion valuation.

Dropbox can credit its success to a lot of hard work, certainly, but Houston and Ferdowsi also stepped into the game at a crucial moment. Houston started coding his software on a bus from Boston to New York in November of 2006. Just two months later Steve Jobs announced the iPhone, a device that would put a computer in the hands people never tempted by a smart phone before. For those who already worked from both a home and a work computer, the iPhone made three. Today over a third of American adults own a smartphone, according to Pew Internet — a number that will only get bigger.

Other things fell into place, too. By the fall of 2008, almost every major computer manufacturer had a netbook on the market for $400 or less. Smartphones made

thumb drives an inadequate solution, and e-mail wasn't set up for the kind of heavy lifting the new computing landscape required. Benjamin Pierce, a computer scientist at the University of Pennsylvania explains, "The need for replication technology has gotten absolutely urgent."

Dropbox found a wide audience just beginning to understand that they required something better. Dropbox snagged 70,000 signups overnight when they posted a video demo to Digg in March 2008. "Arash and I were sitting across from each other doing database queries, and there were hundreds of people signing up per minute," says Houston. "We watched it hit the front page of Digg. Then the top 10. We were thinking, holy shit, what did we do here?"

By their launch six months later, they had 100,000 users had signed up. What had happened was that the first wave started sharing Dropbox files with their coworkers, friends, and family, hooking an extended network into the service. Almost immediately, those who didn't actively search out the solution outnumbered early adopters.

Growth continued to ramp up exponentially. Fifteen months later they had 4 million registered, and 15 after that they had 25 million.

The users had declared a solution.

Drew Houston was running around Cambridge with two years of startup sweat stored on a thumb drive in his pocket. It was 3 gigabytes' worth of planning and architecture and strategy for an SAT prep service that he started while he was still a computer science student at MIT. It was late nights of reading how-to books on the roof of his fraternity and early mornings coding in his boxers with a friend in the then-Cambridge-based Y Combinator.

Once he'd almost left it at Starbucks. Another time he yanked it from his pocket to find that he'd bent the connector. If it weren't for a flash of panic as he approached the door or a careful bit of manual coaxing, all his work would have vanished with the integrity of the object itself. He was in a precarious situation — basically one spin in the washing machine away from losing his company.

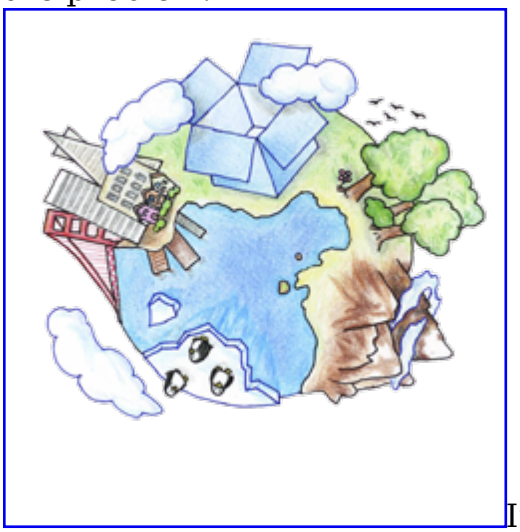
It was a particularly vexing situation for Houston because it was a new one. While at MIT, his stuff was always available to him through Athena, the campus's so-called distributed computing system. All he had to do was log in to any connected workstation (his own included) and every one of his papers and proofs would be there waiting for him. No loading files onto an external thingamabob or e-mailing documents to himself, MIT's goddess of wisdom promised take care of his.

But with a cap and a diploma in 2006, his access to everything from everywhere was gone. Even though Athena was sprouted from a system developed 20 years prior at Carnegie Mellon, it was tied to an institution — and one he was no longer a part of. While Houston was at MIT he had cobbled together some code that would handle his backup needs, but the quick fix wasn't sturdy enough to handle, by this point, years of his work. His best option became physically hauling files from one location to another on a plastic and metal stick — a string tied around the finger as a reminder that there should be a better way.

Fed up with the burden, Houston did what most of us would have done. He Googled it. What he found was a grab bag of semi-solutions. “There were the backup guys. There were other services that let you upload files to a website. There were other tools to sync things across your computers. There were other tools to share files or send big files. What do people use for this? It’s a hundred different things.” And none met his fledgling company’s needs.

At the same time Houston was getting antsy for another reason; despite taking a year off from MIT to work on his SAT prep service and then taking it on like a second full-time job after he graduated, the startup was lacking traction. Y Combinator wasn’t interested in funding it. And that friend that he was coding with into the wee hours of the morning? “He got his startup further off the ground in four months than I had in a couple of years.”

Houston’s earliest memory of job lust was wanting to own a business. He had been coding since he was five. Perhaps, he thought, he could hack his way out of the problem.



In the spring of 2007, after three months of intense focus and fervent coding, Houston submitted his second application to Y Combinator. But there was only one problem: He didn’t have a co-founder — a mandatory requirement for gaining entry to the boot camp and funding that would surely follow.

Unbeknown to Houston, a third-year at MIT named Arash Ferdowsi was also antsy to put his coding skills to work on something big. A good friend of his named Kyle Vogt had recently quit MIT, moved to California, and was having tons of success with a video streaming service called Justin.tv. Justin.tv was covered in the *San Francisco Chronicle* and invited onto the *Today* show. Meanwhile Ferdowsi was sitting in his MIT dorm room taking a distracted stab at a problem set and wondering what the hell he was doing there.

Cloud-like software trickled to market in the ’90s and 2000s, but these tools failed because they were either too complicated to use, tied to a single platform, or both

Ferdowsi wanted to get into the details of what happens when services break big — shooting up to tens or hundreds of millions of users. “I was fascinated by the

scale that Facebook was operating at, and similarly the things that Kyle was doing at Justin.tv." So he filled schedule with classes about the database management and "making things like attributed systems work well." He'd also lined up an internship at the world's largest social network for the summer after his junior year.

Houston was also watching Vogt's impressive Silicon Valley climb. The two had met a couple of years earlier in an entrepreneur club on campus. When word got out that Houston was in need of a partner, Vogt connected the dots. Houston brought the initial idea, a mockup of the service, and the business experience, and Ferdowsi had the desire and interest in the backend to tackle a problem of scale that had eluded services past.

The match took, and the pair walked into their Y Combinator interview only having met for the first time two weeks prior.

The interview was a slam dunk, largely because Houston's application to the program had laid out a strikingly clear vision for what Dropbox would become — an all-in-one synchronization, backup and collaboration tool that "just works." In fact, it's now the stuff of tech entrepreneur lore, one of the folded and coffee-stained printouts that young founders pin to cheap bulletin boards.

Y Combinator's 2007 demo day delivered Dropbox $1.2 million in funding. Ferdowsi dropped out of MIT with only one semester left until graduation. Now with money and time, the hard work began. They had to find a solution to a problem that had resisted one for decades.

In 1982, a computer science professor at Carnegie Mellon University named Jim Morris wrote a manifesto about connectivity and collaboration. "The primary goal," he wrote, "should be to broaden, deepen, and improve the communication among students." The way to do that was to create a computing environment where information could be shared. So together with IBM, Morris set out to create the kind of network at Carnegie Mellon he predicted would be ubiquitous in the future. By tethering $3,000 workstations together with IBM's state-of-the-art wiring, Morris created the Andrew Project, and within it the Andrew File System, or AFS. AFS was the first safe and efficient distributed computing system, available to both students and professors on campus.

It was a clear precursor to the Dropbox-like software packages today. First, it was a big step toward getting your files to be everywhere at once. Though files were stored on central servers, AFS created the illusion for the user that they were local. The first time a workstation wanted access to an AFS file, the computer would go fetch it from the server. But once retrieved, the computer would cache the file on its local disk, so the next time someone wanted to pull it up from that location, the file would be available locally.

Still, a decade later, students complained that the system was too slow. Part of the problem was that files were sent from server to client and back to the server whole. Every time a file needed updating, even if only a single word had changed, the entire document was re-sent — and sent over a maddeningly slow Ethernet connection. By 1996, Carnegie Mellon had figured that sending only

relevant chunks back and forth would save time and bandwidth, but communication speeds had yet to make that change truly valuable.

There were also issues with the way the Andrew File System got along with the software. AFS rendered some Unix applications, like the mail delivery system Sendmail, unreliable. The computer's applications "were not written in the expectation that basic file system operations or close system calls could fail," wrote Nathaniel Borenstein, one of Andrew's principle designers. And when they did fail, the programs couldn't rebound.

Nonetheless, AFS made massive leaps in distributed computing. Over the course of its 30 years, it was tweaked and streamlined and revamped a hundred times over. MIT picked it up for its distributed computing network not long after AFS got up and running. MIT's version, called Athena, allowed students (like Drew Houston and Arash Ferdowsi) access to all their stuff from any connected computer.

Good for those institutions, but not for the rest of us. Until recently, distributed computer clusters have huddled around a central server managed by an organization. Companies and universities maintained their own server space and dedicated IT team to manage it. That works well internally, but getting files to a machine outside the network turned basic computing into a key-fob initiated, slow-motion train wreck.

Decade depending, the floppy disk, CD or thumb drive were tasked with picking up the slack. They didn't provide the most elegant fix, but they were an effective way to port data from one station to the next. But for figuring out what version of a document was last updated and by whom, you were on your own.

University-led programs like the Coda File System out of Carnegie Mellon in 1987 aimed for something not so dependent on a single organization. Because Coda was open source, it could be pushed beyond the work of a single design team. Coda was kind with bandwidth before broadband made the client-to-server connection easier, and it was also generous with offline access, providing cached copies of files when the Internet was out of reach. When disagreements between files occurred — a problem that has such deep roots it took a half-hour conversation with Mahadev Satyanarayanan, Coda's project lead, to explain — issues were mediated with a set of application-specific plug-ins, some acrobatic hand waving, and, ultimately, some damage control that looped in the user.

But in the end, Coda was never commercialized. It wasn't a light piece of software that could be sent to the masses. It required substantial work on the users' part to get up and running. Satyanarayanan admits the system is "complex both to implement and for the user to think about." And besides, his job is to lay out a high level proof of concept, not to dig into gritty startup details.

Still, the average computer user was left without an effective way of getting their work from one computer to the next. Software finally aimed at that audience finally trickled to market in the '90s and 2000s, but these tools failed because they were either too complicated to use, tied to a single platform, or both.

Unison, a manual syncing tool developed by Benjamin Pierce at the University of

Pennsylvania, was one of the first cross-platform services. “Unison goes through an enormous amount of trouble to do the right thing cross platform,” explains Pierce, “but it’s extremely hard.” The difficulty showed in the software’s interactions with the operating systems. And it didn’t function in the background, quietly coordinating matching files. If you wanted the software to do its job, you had to prod it to do so. Unison, though on somewhere between 100,000 and a million computers, still needed to be more user friendly to edge out the humble thumb drive.

Commercial applications even five years ago showed some serious character flaws. Halfhearted attempts by Apple and Microsoft refused to cooperate with other operating systems, so Mac users at home who sat at a PC at work were out of luck. And issues with reliability and accessibility blocked widespread adoption. Other programs were just too clunky and slow to be worth the effort. Backup programs like XDrive were not able to cross the space from client to server gracefully. Their systems transferred data chunk by chunk every time you needed it — something akin to moving across country by driving one box across at a time. The application’s protocols — the way the computer talked to the server — were caught off guard by this coast-to-coast trip. The sluggish pace tripped up operating systems not expecting the lag.

A handful of issues made the user experience, in all cases, unbearable. It was high time for someone to swoop in and do syncing right.

Houston can tell the story of his service with any number of creatively mixed metaphors — proof he’s still stage testing the simplest way to tell people what Dropbox does and how it got to be so big. Ferdowsi, though, is better at recalling all the stuff that went down behind the scenes because he still exists there. Houston has become the voice of the company — something he had been rehearsing for on his Phi Delta Theta roof years earlier — while Ferdowsi bounces from meeting to meeting, firmly planted on the engineering side.

They’re working to move Dropbox forward, which is tricky when Dropbox’s success can be largely attributed to the fact that it’s pretty feature free. It was launched in the Fall of 2008 as what we’re now calling a “lean startup” — a basic service sent out to the masses to be stage tested and tweaked on the go. The enthusiasm for the release was startling. It started with a simple box that users could drop their stuff into so people can access their music, pictures, and

documents online and at every connected workstation.

The plan was always to add more features. When early requests revolved around the ability to turn any folder on your computer into a Dropbox folder, the founders resisted. “That sounds like a cool idea but it’s really hard to pull off in a way that’s easy for people to understand,” explains Ferdowsi, who established his status as the gatekeeper of Dropbox’s engineering pragmatism early on. “You can imagine installing Dropbox on a home computer and then installing it at work and then all of a sudden your work documents get merged with home documents.” So the founders resisted.

Y Combinator’s 2007 demo day delivered Dropbox $1.2 million in funding. Ferdowsi dropped out of MIT with only one semester left until graduation. Now with money and time, the hard work began.

One reason they decided to keep it simple was that people had a hard time understanding even the simplest parts of their service. Just a few months after their initial launch, the Dropbox team decided to conduct its own usability tests. They dragged a mic — usually clamped to some piece of office music equipment — into their conference room and had people they’d recruited from Craigslist try out the program. On a connected computer stationed in another room they anxiously watched what went down. “The whole team sat there as people bumbled their way through the installation process,” says Houston. “They couldn’t even get the damn thing on the computer. For everyone in the room, it was an extremely painful experience.”

It was a tough moment for the team, but it just ended up confirming what they’d learned from their own experiences: Simplicity is key.

The founders’ vision for Dropbox was that it work so quickly and seamlessly and without fanfare that the user barely notice it at all. If your computer rejects software because it’s not compatible, the service is not working for you. When files go missing after an upload, the service is not working for you. When your computer slows to process a request, the software responsible is not working for you. Sync programs before had all floundered because they didn’t nail one or all of the above. Dropbox learned from the mistakes of others. “What we’ve always wanted to do is keep finding ways to make Dropbox easier to use,” says Ferdowsi.

To accomplish this, Dropbox needed to be on every computer. So when it jumped out of beta and into the mainstream, Houston and Ferdowsi added a Linux client to the existing Mac and Windows version. It may seem an obvious move to net most users, but scores of companies before then had been deterred. The reason is that cross-platform integration is a bitch. Benjamin Pierce, who attempted it for his own sync software, offers a tiny example of the conflicting system slog that starts the moment you hit save. When you save, a record of that action is logged in seconds. But there’s a catch. Macs will note the save the second you ask it to, but PCs are only accurate to every other second. “So now imagine trying to write a program that is checking whether the file has the same modification time on all the machines that it lives on,” explains Pierce. “You have to realize, wait a minute, this is a Windows machine, so I shouldn’t be checking if it’s the same

modification time, I should be checking whether it's within two seconds."

What makes the process tougher is that obstacles like these are not documented in some Kindle-available book called Cross-Platform 101; issues are uncovered only when engineers smack into them. "This sounds kind of stupid and trivial, and any particular example like this is, but the point is that there are a huge number of these things," says Pierce. Dropbox was dedicated to the discovery.

But cross platform wasn't just an error-prone landmine on the backend, it also needed to be visually consistent on screen. When someone installs Dropbox, the founders wanted it to look as if the service were baked into the operating system from the beginning. Windows and Linux made this easy with an API that allows programmers to integrate in their own icons. Apple, on the other hand, made it obvious — even to their own MobileMe team — that they didn't really want anyone mucking around in their Finder.

MobileMe's failure to integrate turned out to be an issue that ultimately buried it. E-mail outages and sluggish syncing plagued the service. Its sidelined status within the Apple was such a widely regarded misstep that even Steve Jobs admitted, "it wasn't our finest hour."

Dropbox didn't let Apple's resistance to play with others deter them, and eventually they were able to hack their way in. Sure, Dropbox didn't need a highlighting system in the Finder. They didn't need everything to look the same cross platform, either. But to do both meant a cohesive package without asking the user to take any leap in understanding.

It was with that goal in mind that they approached every roadblock — even the ones covered in decades of dissertations and published papers. As every computer scientist in this space will attest, when two people are working on the same document, syncing the versions can get thorny. What happens when tweaks made in tandem force two versions of the same file to disagree? Dropbox decided to handle conflicts like this: If two people update the same file at the same time, Dropbox kicks one version below the other for the user to deal with. The founders knew that losing work would be a deal breaker for most users. They didn't want to lose any data ever, and with this solution, they hadn't.

You could almost hear the forehead slapping when Unison's Pierce describes how intuitive Dropbox's solution is: "I think that's probably a very clever decision on their part because that kind of conflict for most people is rare. And then you haven't bothered to make people learn something in order to use the tool. You can just use the tool and be happy." There's no special alert to bother the user or software-controlled choice between copies, which could erase work. The action that Dropbox requires by a human — a decision about what version to accept — makes for about as little an intrusion as they come.

Dropbox also chipped away at seconds in the same methodical they tackled the problems before it. If everything just works, there should be no aggravated finger tapping as we wait for filed to upload. They were always wary of an XDrive repeat pushing away users, so they spent hours upon hours in their first few years training the service to work faster.

One thing they did was engineer the service to tread lightly on bandwidth and on the computer's operating system — sending as few updates as possible across the client-server divide — which shaves seconds and computing cycles. They do it with a sort of CSI-type approach. The program takes a numerical fingerprint of everything in the folder. If something shifts within a file — something's moved or renamed or even a single letter changes — Dropbox compares the new fingerprint to the old one. "In the event that you have a large Photoshop file and you make one change, Dropbox picks out the part of the file that changes and just uploads that," explains Ferdowsi. "The OS will tell you that one file got deleted and another got added, but we were smarter in noticing that the file is already there."

They were also smarter about handling the distance between computer and server. Instead of that one-box-at-a-time, cross-country haul of programs past, Dropbox created a custom protocol that bundles and compresses files into groups, sending them over in batches. "The protocol was designed differently from what everyone else was doing," says Ferdowsi. Moving items efficiently shaves seconds from the server to client call — the foundation of their business.

They've also taken a load off their servers by trimming them of unnecessary excess. Throw the new Beyoncé single into your Dropbox and the service will upload the song once. But say 1,000 other users are digging the same track, instead of weighing down the back end with lots of repeats, Dropbox practices deduplication. An algorithm sniffs out copies and kicks the extras from the system so they're only storing one version of a file. This is just one glimpse into a reorganized back end, which doesn't just replicate the tree of files on your computer; it splits up and thins out the data to make it easier to retrieve.

But as the service matured, more users trusting it to sync swiftly and securely, they needed to unburden their servers and the users' operating systems. Instead letting the program work to sniff out changes, Dropbox made the changes come to them. The move slashed a ton of time and computing energy. "If you want Dropbox to be pretty responsive, we would probably be checking at least every minute. I guess that would be thousands of times per day per client," explains Ferdowsi, which adds up to a lot of unnecessary bandwidth. The change to a fully event-driven system, by Ferdowsi's back-of-the-envelope calculation, sped up the service by three orders of magnitude.

"I guess we're all sort of very obsessed with milliseconds everywhere," says Ferdowsi, matter of factly. By pulling together a lot of little things, trimming any excesses, and making sure all the solutions were glued together in a way that puts the user first, Dropbox discovered a broad audience. But now that they've captured the attention of millions, the key is to figure out how to hold onto it.

Both Houston and Ferdowsi bristle at the notion that they might have solved the data-sync problem. They're proud of their service, and they know it works well, but there's still a lot more they want to do.

To move forward, they're bringing on new engineers like crazy — expecting to jump from 70 to 200 people on staff by the end of next year. It means their

current top-floor Market Street digs are crowded with giant glowing monitors, but not for long. They plan to move to a new space in San Francisco's China Basin at the start of the new year. To top it all off, they've also just gained $250 million of series B funding led by Index Ventures, an impressive amount that will no doubt help with the expansion.

But really, you get the feeling that they're ramping up quickly in preparation for a nasty fight. And with iCloud, Apple's thrown the first major punch. Sure, Apple has stumbled in the space before, but now with a few more years of development under its belt, they're in a good position to snatch the "it just works" catch phrase right from the Dropbox founders' mouths.

There are others gunning for Dropbox, too. A slew of similar start-ups hope to take a slice of their market share, and a Google release, Houston worries, could lure away a significant chunk of users.

In the meantime, Dropbox is trying to make their appeal even stronger. A new feature roll out early in the Spring to make sharing files even easier. With just a link, a gallery of photos will open up in its own webpage. People will no longer have to sign up for the service to access content sent to them by registered Dropbox users.

They also hope to move beyond folders. Eventually Dropbox will automatically split photos into albums based on camera type and event time — something like what iPhoto already does photos dumped from camera to computer. The goal is to save the time users spend organizing their Dropbox into a hierarchy of folders. Automatic uploads, something iCloud is already doing, is also on their list of to dos.

And even further in the future, they hope to expand the pool of devices that can talk to each other — basically eliminating your computer as the middle man. A point-and-shoot, they think, could ship pictures directly to your television and family for big-screen viewing — even before you board the flight back home.

"Soon you'll walk into a Best Buy or Fry's and you'll see that little box everywhere," says Houston. "This is the first time I will see my photos or play my music in my living room and it's not going to be a science project."

The more devices with a little box on them, Dropbox hopes, the harder it will be for others to catch up. But it's a gamble. They're gearing up for a massive feature add when the service has so long worked for people because it's stayed simple. The competition is right on their tail. May the best solution win.

Photo: Arash Ferdowsi and Drew Houston. Courtesy DropBox

Drawings: Hand-drawn artwork seen around Dropbox. Courtesy Dropbox

Pages: 1 2 View All

Previously

Around the Web

You Might Like

Related Links by Contextly


Amazon, Dropbox, Google and You Win in Cloud-Music Copyright Decision


4th Time a Charm for Apple? From iDisk to .Mac to MobileMe to iCloud


Google Docs Becomes Google ‘Any File’ as Cloud Wars Heat Up


The 7 Coolest Startups You Haven’t Heard of Yet


From iCloud to Dropbox: 5 Cloud Services Compared


Dropbox - Simplify your life


Dec. 21, 1898: The Curies Discover Radium


Fight or Slight: This Heavy Bag Hits Back


#40dollars And No Mule: The White House and Social Media Strike Back


Half A Billion Dollars: Why Apple's Acquisition Of Anobit Matters


Mozilla + Google: Browser Wars Are Really Search Wars by Proxy

Rachel Swaby is a freelance writer living in San Francisco. She's a frequent contributor to Wired, Gizmodo and Afar.
Follow @rachelswaby on Twitter.

Post Comment | 11 Comments and 120 Reactions | Permalink

Back to top

Like 1 person liked this.

Login

Add New Comment

Type your comment here.

Sort by popular now

Showing 11 comments

moonbasedelta

Comparing Dropbox and iCloud is like comparing oranges and (A)pples. Just sensationalism with a headline. An interesting creation story on dropbox, though. One thing, if we do make any comparisons, is that Dropbox will run on any platform and any OS. iCloud won't run on anything but Lion (10.7) on the Mac side of things, and has requirements for other platforms. Dropbox wins.

Whereas Dropbox has made their technology available to EVERYONE including the dumbed-down masses, Apple is trying to do the same but continuing to fail, Instead of being inclusive, they have become exclusive to some degree. I can't use iCloud if I'm running Mac OS 10.6? And there's many reasons why I don't want to use Lion, and even iCloud actually. As a MobileMe user and a Mac user of more than 20 years, all I can say is Apple sure knows how to alienate their long standing customers and their base of power users, people who have supported the company for decades—all in the name of dumbing down technology for the masses. iCloud should be called iCould.

2 hours ago 2 Likes Like Reply

uther pendragon

Brilliant implementation of concept. 3 thumbs up. (Don't ask.)

2 hours ago 1 Like Like Reply

The Real Mxyzptlk

Ugh. Someone didn't pay attention during English Composition I.

"Backup programs like XDrive, was not able to cross the space from client to server gracefully. Their systems transferred data chunk by chunk every time you needed it..."

3 hours ago 1 Like Like Reply

Marc Brodeur

Yea, I counted at least 5 obvious typos or errors. It was a good story, though.

1 hour ago in reply to The Real Mxyzptlk Like Reply

rawitten

The article states Jobs announced the iPad in 2006. Should say iPhone.

4 hours ago 1 Like Like Reply

TheGreeks

All this time I thought drop box was just a nice wrapper around rsync ... and now it turns out that they solved one of the fundamental challenges of computer science.

1 hour ago Like Reply

T.A. (Tim) Walker

"iCloud may be cross platform averse, but..."

If you're wondering why I'll be sticking with Dropbox until at least when (if?) some outfit like Apple Borgs them, it's right there in that sentence. Dropbox gives me a common folder on our Mac and Linux machines, which I can use to sync not just files, but configs, encrypted stuff, my Calibre (ebook/Kindle) library and just about everything else I want to have access to on any computer I'm using.

If/when iCloud can work on non-Macs (and especially on Linux distros that aren't Ubuntu) as well and as seamlessly as Dropbox, then we'll talk... otherwise, I'm staying put, thanks.

1 hour ago Like Reply

omega_factor

Why would you give your files and data to someone?

Cloud = Big Brother!

Oh, and wasn't Dropbox's so encryption recently confirmed to not be enrypted at their end? Perhaps it's easier to let uncle sam inspect your data that way!?

1 hour ago Like Reply

uther pendragon

There are many available tools to encrypt your data, PGP for instance. If you feel the need, handle the encryption yourself.

1 hour ago in reply to omega_factor Like Reply

T.A. (Tim) Walker

Already do :-)

35 minutes ago in reply to uther pendragon Like Reply

omega_factor

I do!

1 hour ago in reply to uther pendragon Like Reply

M Subscribe by email S RSS

Reactions