Oral History of Andreas “Andy” Bechtolsheim

Interviewed by:
Douglas Fairbairn

Recorded: July 17, 2015
Mountain View, California

CHM Reference number: X7546.2016

Doug Fairbairn: All right. This is the July 17, 2015. I'm Doug Fairbairn, along with Uday Kapoor. And we're here to interview Andreas Bechtolsheim, otherwise known as Andy Bechtolsheim, who has had a long and extensive career in Silicon Valley, and we'd like delve into the details of that, but also go back to the very beginning in terms of-- I understand you grew up and you were born and grew up in southern Germany-- and so we'd actually like to start the interview back then. I think you were-- well, tell me about the environment. It was not a technical environment that you grew up in. Is that correct?

Andy Bechtolsheim: Yes. I grew up on a farm in Bavaria. My dad was an elementary school teacher and my mom was a housewife. We didn't run the farm, it was leased to a farmer, but it was great environment. The farm was a mile away from the next village and we couldn't see another house from our house. We looked out on beautiful meadows, forests and the Alps in the distance.

Fairbairn: Sort of idyllic scene.

Bechtolsheim: Yes the nature was idyllic. There wasn't much interaction with other people, but that did not bother me since I was mostly interested in understanding how things worked. As early as I could remember, like four or five, I started taking things apart and putting them back together just to see what was inside.

Fairbairn: Now what kinds of things were those?

Bechtolsheim: Well, my dad had a tape recorder, one of the old reel models, and he was getting kind of worried as I was taking it apart, but I put it back together and it still worked. One funny story about my early life is that I did believe in Santa Claus, but I wanted to make sure that Santa Claus would get me the right stuff, so I literally sent Santa Claus a wish list letter with the exact SKU numbers of the experimental kits that I wanted. Magically they showed up under the Christmas tree.

Fairbairn: Leave nothing to error, nothing to chance.

Bechtolsheim: Yes, one has to plan ahead. At age six I was building little electric motors from wires and magnets..It does create this deep sense of wonder, how does it work? Why does the needle of the compass always point north? How can phosphor paper glow in the dark without needing a battery?

These early experiments I think motivated me to dig deeper and deeper into how things actually worked. So when I got into my teen years--

Fairbairn: Just to sort of set the stage, you were born in '55?

Bechtolsheim: Yes, I'm born in '55. So in the early '60s, I was doing just basic electric experiments, not even electronics yet. The electronics started in the late '60s, when there were transistors. Building stuff with vacuum tube was kind of challenging, but with transistors you could make a lot of neat little things. At that point, the basement of my parents' house turned into my hobby department, and I spent all my free time just building stuff. One of my favorite projects was a frequency counter that also allowed time measurement at the millisecond level. It had plasma display tubes and it worked like a charm.

Fairbairn: So these are plasma numeric displays?

Bechtolsheim: Yeah, it required TTL logic, which at the time was available from Texas Instruments, and it all worked just beautifully. This device became suddenly very popular with the local car racing club because they needed somebody to time the races. So I did a light gate measurement device so that when the car went through, we could measure exactly to the millisecond how fast it was.

Fairbairn: Now, before you get any further, was there anybody else in your family that had this same inclination or were you sort of the only one--

Bechtolsheim: I am the only person in my extended family that ever did anything engineering-related, with the exception of my great-great-uncle who invented the milk centrifuge in the late 1800s. He called it the alpha centrifuge and he sold his patent to a Swedish company called Laval. Many years later they changed the name of the Swedish company to Alfa Laval, because they're still the world leader in milk centrifuges today. But that great-great uncle was the only person in my family that had a career in the field of engineering.

Fairbairn: OK, so in Germany, before the internet, how did you get parts? I mean, where did you even become aware of these things?

Bechtolsheim: Well, I had one advantage. In the little village where we lived at that time, there was a small local electronics company that was building controllers for numerically controlled machines. I became an intern at this company to help them build new products. Part of the deal was I could build whatever I wanted.

This then led to the design of my first microprocessor system. So now we're in the early '70s, and I have to say I feel extremely lucky that I was a teenager when the first Intel microprocessor arrived. The Intel 4004, of course, was kind of a toy processor, and the 8008 was not much better. But the first Intel 8080 microprocessor actually made sense. It was fast enough to control a metal manufacturing machine. So in 1973, when the first 8080 chips arrived in Germany, I convinced this local electronics company who previously built controllers based on digital logic that were unique for every machine that they could build a microprocessor based platform that would work for any kind of machine and program it.

Fairbairn: OK, so where did you even get the idea of programming. Was there--

Bechtolsheim: Well, there were no software people in this company. Nobody knew what programming meant, and neither did I. So I boldly proclaimed I would just build them this system including the software. I worked day and night on this project because it was so fascinating. I was always curious how computers worked, and there is no better way to learn it than building your own computer from the bottom up.

Initially, the programs were all running out of SRAM with battery backup because I didn't have any other way to load the program than keying in the binary code one command at a time. But when you tripped a bug in the code, it would overflow the stack, and overwrite all the stuff you just coded. So the first thing I added was a memory protect switch, so you could actually make the memory read-only. I programmed this thing in binary code because there was no assemblers or compilers. So I actually coded this system with a hex-binary keyboard. Each 8-bit instruction took two letters to type.

Fairbairn: And this-- you learned this completely on your own. There weren't any teachers or

Bechtolsheim: Exactly. There was nothing like building your first computer system on your own and then programming it. I had to write a little operating system and routines to do, advanced math, because this machine was supposed to be able to do rotary motions, and that required a floating point library. So I did floating point on this incredible slow 8-bit CPU. It was almost hilarious because while you can do anything on a computer, the advantage of these numerically-controlled machines was that they moved at a fairly slow pace, so the 8080 was a perfect speed match for this particular application.

And because I was not even 16, they couldn't make me an employee of the company, so the owner decided he would just pay me the royalty for each box they would sell. That worked out really well to the point where I suddenly was making more money than my dad as a teacher.

Fairbairn: I was going to ask you, how did you--

Bechtolsheim: So that kind of shaped my future expectation that I really want to just be in the business of selling stuff, and collect a royalty. It was a key moment, because you don't really learn in school what you're going to do one day. You have to figure it out yourself, based on your own experience--

Uday Kapoor: I was gonna ask, so in school, did you want to be in sports as well, or were you--

Bechtolsheim: Sports was my least favorite subject. I was really good at physics, math, biochemistry, and was desperate to learn about computers but there was no computer or computer education at the school at the time. And none of the other students even knew what a computer was. They had never seen one. For me, it was the most interesting thing in the world but I could not share it with anyone. There was one other student who was two years older than me who eventually joined this company and ended up working there. But it was just the two of us kept working on this system, and he stayed behind when I left to come to the US.

Fairbairn: So you went through, what, the equivalent of high school or whatever living in that same area right?

Bechtolsheim: Yes. I finished high school in '73, and shortly afterwards, Intel introduced the MDS-80 development system. So you could actually buy a box that had an assembler although not a compiler. So we got a couple of those systems. It did made it much easier to program.

At that point, I really wanted to study computers. After high-school, I spent two years at the Technical University in Munich, except they didn't have any computers I could get access to either. It was a heartbreaking experience because you can only really learn programming by doing it. You can't just go into a classroom with lots of people and see the professor scribbling on the wall. So, one year into this, I decided this wasn't working for me, so I better--

Fairbairn: So what year was that?

Bechtolsheim: This was '74. I finished high school in '73, and was enrolled at the University in Munich until ’75, even though I spent most of my time further developing the microprocessor system at the electronics company in my home town. In the meanwhile, I participated in the German science fair competition called “Jugend Forscht", which is a nationwide event for students to come up with interesting experiments, and in 1974, I won the first prize in physics for a very cute experiment, which was how to measure fluid flow non-intrusively with ultrasonic waves. This required a fair amount of engineering that I build custom for this project. After I won the German Science Fair competition, it made it much easier to get a scholarship to come to the US to study.

I received a Fulbright scholarship that allowed me to study Computer Engineering in the States. I arrived here in 1975 and I got my master degree from Carnegie Mellon in 1976. I was only 20 at the time, because I kind of skipped undergraduate school. I will be forever grateful to the admission officer at Carnegie Mellon who let me go into grad school, even though I didn't have an undergraduate degree.

Fairbairn: So you didn't have to take all the preliminary--

Bechtolsheim: It was a funny conversation because I had barely taken any classes in Germany since I spent most of my time working at the electronics company. So, he asked me, how do you know you can master the graduate program. I said, well, I can always take more classes. But it wasn't necessary.

Fairbairn: So what got you to Carnegie Mellon?

Bechtolsheim: Well, actually, I didn't know where to go, because the US has hundreds of schools, including many good computer science schools. It was actually the Fulbright people who suggested that. Carnegie Mellon had a great computer science program. And they had high level of funding from ARPA . They were working on speech recognition and robotics and multi-processing systems, which was the project that I got involved with. I loved the fact that they not only had classes but also put theory into practice by building very ambitious projects.

Fairbairn: C.mmp?

Bechtolsheim: C.mmp was before me. I was working on the Cm* project, which was the newer version. Digital Equipment donated some LSI 11 logic boards, and so I had an 16-bit micro computer to start with. The first problem was addressing space. You couldn't even address all the shared memory with 16-bit, but there was a way to do some memory extensions.

I was responsible for designing and building the network interface for Cm* so they actually could connect together. This is before Ethernet, but I managed to built something that actually worked quite well. It was a low latency fabric that supported remote memory accesses and mailboxes.

Fairbairn: So by that time, you had both computer science and networking experience, right?

Bechtolsheim: Yes, networking was my Master project, and funny enough, here I am 40 years later still working on data center networking. Same topic, how to interface all these computers except the scale is vastly bigger. After I got my Master’s degree in 1976, I spent another year at Carnegie Mellon, but I really wanted to come to Silicon Valley since this is where all the action was, even back then.

Fairbairn: So-- OK. Before you get there, were your parents very supportive of this? I mean, were they happy to have you going out to the United States or--

Bechtolsheim: Well, they always supported me in my interests, and I think they realized that eventually I would end up living here.

Fairbairn: And you were making money?

Bechtolsheim: Not as a student. I was supported by scholarships. My next career goal was to come to Silicon Valley. So in early 1977 I interviewed with Intel Corporation for a summer job. Justin Rattner, who is still with Intel today, offered me an intern job, so I drove out here. I arrived in May of 1977 only to find out that Justin was just transferred to the new Intel facility in Oregon. Justin told me I could join him there, but I didn't want to go to Oregon. I wanted to be in Silicon Valley. So I was here with nothing to do. What I did not realize at the time was this was probably the best thing that could have happened to me.

I was staying with a friend who was a computer operator at Stanford at the time. He showed me around Stanford and there was a sign on the bulletin board from a professor looking for help with computer-aided design programming. I was familiar with CAD, because I had used CAD for my Cm* network design, So I applied for that job and did that for the summer.

At the end of the summer, the professor said: why don't you just stay here? You have a master’s degree, we can just transfer you. So while I never applied to Stanford, I ended up there by sheer coincidence because my summer job with Intel did not work out as planned.

Stanford, as you know, is a wonderful place. I started in the PhD program in the fall of 1977. In '78, I had the great fortune to become a no-fee consultant at Xerox PARC, where I met you. A no-fee consultant meant they didn't pay me anything, but I could hang out there as much as I wanted. This was like a dream. Pretty soon, I was hanging out at PARC all the time because they had this incredible environment with Alto computers networked together with laser printers that nobody else in the world had ever seen. It was the most amazing place.

Fairbairn: Now, who got you over to Stanford in the first place?

Bechtolsheim: It was Bill Van Cleemput. And at Xerox PARC, my sponsor was Lynn Conway who wanted to have students try out her new chip design system she was working on. I did a chip design using those tools. The chip even taped out on a shuttle, and it kind of worked.

It was kind of a weird chip. It did logarithmic arithmetic, instead of linear arithmetic, with all numbers expressed as logarithms, which makes multiplication an add, which is great-- but, of course, makes

addition and subtraction kind of a funny function. You have to do this in tables, and it was an interesting project, even though there was no commercial aspect to this.

But what I realized was that the Alto computer, the first true usable personal computer, was the ideal environment for doing chip design, and I thought every scientist and engineer needed one of these, because they could be so much more productive, rather than having a VT50 terminal hooked up to a VAX mini-computer time-sharing system.

The weird thing about Xerox PARC was that it was a research lab. They came up with all these incredible ideas but were not productizing these things to sell them commercially, which was really odd. So my next thought was if they are not productizing the Alto, maybe I should build my own version at Stanford, based on the Motorola 68000 chip, which was the first 32-bit microprocessor out at the time.

In 1979 we received the first 68000 development board. I built an Ethernet network interface board and a bit-map graphics card that drove a black and white monitor, and here was my first workstation, which eventually became known as the Stanford University Network or SUN workstation.

Fairbairn: Now, did it have sort of the bitmap displays similar to the Alto?

Bechtolsheim: Yes, it has the same kind of bitmap display as the Alto. If I hadn't been to Xerox, I would have never thought of this, so I can't take credit for coming up with the idea. What makes any computer interesting is of course the software that gets written for it. In the early days, we just had some terminal emulator. That's very low level software.

The next idea was that we could actually run Unix on this workstation, meaning Berkeley Unix, although the 68000 didn't have the restart instruction that was required to implement virtual memory. So we had to wait for the next chip to be able to do Virtual Memory. It took Motorola two years to fix it.

On the hardware side, I designed printed circuit boards, so we could manufacture these workstations more easily. By 1981 we were making dozens of boards in the basement of Margaret Jacks Hall.

Fairbairn: So who was funding the development of these?

Bechtolsheim: Well, this was a funny story. The SUN project was supported by DARPA, which paid for my RA-ship and the other people working on the software for this. But then it became clear that there was a significant commercial opportunity here, and Stanford told me that they were a nonprofit institution and were not in the business of building computers beyond the initial prototypes. So we had to spin it out.

My first idea was to license the Sun design to other companies who wanted to build the boards. I signed up seven companies who wanted to build different products using the Sun design.

However none of these companies wanted to build a workstation They all wanted to build multi-user Unix system, like the VAX. They did not understand the potential of a graphics workstation, because they haven't been to Xerox PARC and had no idea what this could do.

I remember having a discussion with Bob Marsh, who was a successful entrepreneur at the time. He had a company called Plexus where I tried to convince them that he should build these workstations. Bob’s argument against it was that there was no software, which was true. There was no software for graphics workstations. My counter argument was, people will write the software! It was completely obvious to me that the developers would come along. But he passed.

Fairbairn: But you were running Unix on ….

Bechtolsheim: We didn't have Unix ported yet. The hardware was designed to run Unix, and at the time, we diddled around with Unix System III, which was the old version, not the Berkeley software distribution. Berkeley 4.2 wasn't finished yet and all that work was happening at Berkeley, not at Stanford.

In the meanwhile, everybody else wanted to build a multi-user computer or various networking devices. Cisco actually licensed the Sun CPU board for their first router. But nobody had this vision to build workstations, and I started to feel that I was going to miss the market opportunity, meaning if I was not going to do this, somebody else would do it.

Fairbairn: By then, Apollo and others were starting to appear right?

Bechtolsheim: Exactly. Apollo was a well-funded company at the time. They had a management team that came from Prime Computers. And they built a huge box with two 68000 chips, in case the first one hit the memory fault, the second would take over and then fetch the page from memory to allow the first chip, which was hung in the suspended state, to continue. This was about the silliest thing I have ever seen to solve the missing virtual memory fault instruction problem is to put two CPUs in, because the second CPU could not really do anything except fix the page fault while the first CPU was hung.

They ended up with a huge box with large boards, with hundreds and hundreds of chips on the boards. It did work, but it was a very expensive box. The Sun workstation design had one quarter of their chips, and much smaller boards, and was clearly much more cost-effective.

At that point, Vinod Khosla, called me up. This was at the end of 1981, and he heard about the licensing operation, but as soon as we started talking it was clear that the right thing was to start a company to build these workstations and sell them in volume. Vinod knew Scott McNealy who was his best friend from business school. Vinod and I wrote up Sun’s first and only five-page business plan. A week later, we meet with two venture firms, and they agree to write us a check, and we had a company. It was actually very quick to that point.

Fairbairn: So who were the VCs?

Bechtolsheim: The first two were US Venture and a company called West Coast Ventures.

Fairbairn: Was Forest Baskett involved?

Bechtolsheim: No. Forest, who was a professor and my advisor at Stanford, stayed at Stanford.

Anyway, at that point, I did drop out of the Stanford PhD program because you can’t start a company and finish your PhD at the same time. However I was still using the Stanford CAD system, because it took us another couple of years to actually get the CAD software running on the Sun workstation.

Fairbairn: What CAD system did you use to build the original Sun?

Bechtolsheim: It was called SUDS, which stood for the Stanford University Drawing System, which was written by a couple of people who were at Stanford before me.

Fairbairn: Dick Helliwell?

Bechtolsheim: Yes, you remember them. They had this goal of building a new PDP-10 computer, but you needed, of course, a CAD system before you could build the computer. So by the time they built the CAD system, they graduated and they ended up working at Digital, I guess. And Digital actually used the same CAD system which they created at Stanford internally.

The SUDS CAD system was only used at Stanford, at Carnegie Mellon, and MIT because it only ran on a PDP-10 and you needed a special keyboard with three meta characters, because all the commands were

keystroke abbreviations. To learn this thing, you had to type for weeks to get really good at this. But once you mastered it, it was quite fast. There was no mouse, you had to do all the commands with keystrokes.

It actually was quite effective, except it only ran on the PDP-10 which was a timeshared computer. So the only time you could use the system was the middle of the night and early morning when there was no one else using the computer. So I had to have these odd working hours. I go to bed really early and then wake up at four. Because then there was nobody else left from the people who worked late nights.

Anyway, without the SUDS CAD system, I could not have built the Sun workstation. Tools are incredibly important in designing machines, and before CAD, people did this with paper and pencil.

Fairbairn: Did it have simulation? Or was it purely--

Bechtolsheim: No, there was no simulation. In fact, we didn't even have static timing checking, and I ended up writing my own timing checking tool that compiled min-max expressions into C. It was a very effective tool. I wish we had that earlier because we had some timing issues in the original design.

Kapoor: I wanted to ask-- of course, this is very intensely technical all the way through-- what about your social life and friend groups?

Bechtolsheim: Social life? I didn't have any social life. I was working day and night on designing new workstations and building the company. That was the only thing that mattered to me at the time.

Kapoor: Right.

Fairbairn: Was there anybody else working with you on this or was this--

Bechtolsheim: Not on the hardware. Since I was the only one who actually knew how this CAD system worked, I was the only one who could do all the schematics. I did hire a young lady to lay out PC boards, so she did some of that. And then there was another fellow student who wrote the boot code that actually would boot the system up. He later joined Digital Equipment.

This is also when I met David Cheriton. He was actually the first user of the Sun workstation because he was writing the software that became known as the V-kernel that became part of the X Windows system. He actually found some bugs in the hardware, because he was the first one who actually used it.

I started several companies with David later in life, but at the time he was a new associate professor at Stanford, and he didn't join Sun. He stayed at Stanford and became a tenured professor.

Sun was incorporated at the end of February 1982, delivered the first production units in May 1982, and Sun was profitable starting the next quarter, because the sales grew so quickly.

Fairbairn: Did you have a business plan?

Bechtolsheim: Yes, the five page business plan that I wrote with Vinod in January.

Fairbairn: Who were you going to sell to?

Bechtolsheim: There were quite a few people who had heard about the Sun project at Stanford and wanted to buy a workstation. So we actually had a backlog of orders before the company even started. Many of the early customers were universities and research institutions who wanted to do software development. What really increased the interest in Sun was when Bill Joy joined a few weeks later and we announced that we would do Berkeley Unix.

Fairbairn: Before you get to that, how much did the hardware cost? What were your cost to build it and how much did you sell it for?

Bechtolsheim: Well, memory was pretty expensive back then, so the initial machine only had 256 Kilobytes, which was not enough to run any interesting software. The first product we shipped actually had one Megabyte, which by today’s standards is a trivial amount. Pretty soon, people wanted two megabytes, and then four and eight and so on.

The one mega-byte Sun workstation with black and white display sold for about $15,000. And the color version was about $25,000. But that was less than half the cost of the Apollo machine at the time.

Fairbairn: So $10,000, $20,000 was the sales price, and they cost a third of that or a quarter of that?

Bechtolsheim: We made about 50% true gross margin, including all the manufacturing overhead. The costs kept coming down with more volume, but people then wanted more memory and more stuff. As a result the prices remained remarkably stable over time. You just got more and more for the same price.

Going back to the beginning of Sun, we needed somebody who really understood software, and while I've done some software in my life, I was not an operating system guru. It was clear that Berkeley Unix was the future. It was still a work in progress, led by Bill Joy at Berkeley who was famous because he was doing most of the development by himself, working day and night.

To convince Bill Joy to join us, we told him that he could hire a bunch of people to help him finish BSD and give it back to Berkeley, since they had the copyright on the code and the contract with DARPA to finish the 4.2 release for the VAX, the same release we wanted on the Sun workstation.
So this was perhaps the first commercial open source project. Bill joined Sun, we hired some really great programmers that helped him to finish the BSD 4.2 release, which was the first version of Unix that had full Virtual memory and a complete networking stack, and we ended up with the same software on the Sun workstation, which of course was much more cost-effective than the VAX.

Fairbairn: So the first Suns were being sold without an OS?

Bechtolsheim: No, they ran Unix level seven. A horrible old Version of UNIX. We told people that we were going to upgrade them to Berkeley Unix, and they actually believed us, because Bill was at Sun. So people bought machines expecting we would upgrade them to the Berkeley Unix operating system six months later.

Not only did we deliver on that but we also upgraded the hardware to the new Motorola 68010 CPU which had the virtual memory instruction, and then we actually had full virtual memory on this fairly, inexpensive computer, which was almost as fast as the VAX780, which cost more than $100,000 at the time. So it was a complete no-brainer in the technical market to buy a Sun workstation running the same software as the VAX computer which cost many times more.

Digital didn't quite believe at the time that this was happening. I remember they had a large customer, Schlumberger, that wanted to stay with the VAX because they had bought a lot of VAXes. They wrote a letter to Ken Olson, the CEO of DEC, saying if they would build them a VAX workstation, they would buy 1,000 of them. And DEC refused to do it. They just didn't believe the market was real. They didn't want to lower the price per computer. They believed in multi-user time sharing. At Sun, we believed in one workstation for each user. Instead of a VT100 terminal, sell them a workstation.

For engineers that are redesigning chips, you needed a workstation. Even for software development, you needed a thing where you could compile quickly. You needed an environment that had more cycles than being stuck on the multi-user system. So it took off very, very quickly in the development community.

Fairbairn: How much money did you raise?

Bechtolsheim: Well, that's a funny story. On the very first round the two VCs committed to give us $2.5 million for roughly half the company, but because we had just met, they identified a risk factor here, which is the founding team met a week earlier, and perhaps we were not going to get along.

So they gave us only a check for $250,000, which was enough for the first three months. And then, if we met a list of milestones they would give us the rest of the money. So we started with only $250,000, but that was actually in our favor because three months later, we came back and said, we are doing much better than we told you, and we actually think we are worth twice as much.

They said, no problem, if you find somebody else who is willing to pay more, we will participate. At that point, we called John Doerr from Kleiner Perkins and Dave Maquardt from August Capital, which I had met at Stanford even before the company was incorporated. In the end we raised almost $5M for half the company, which was very helpful, because we needed the money to grow. We had our own manufacturing, we had to pay for the parts.

Fairbairn: So where did you start? Where was the first building?

Bechtolsheim: The first building was on Walsh Avenue in Santa Clara. We were there for nine months and then we moved to Mountain View. Sun leased many buildings there. Some of them today are Google buildings. Many years later, Sun built a new campus in Menlo Park, which today is the Facebook campus, and another one in Santa Clara, where most of the Sun-Oracle division is today.

Fairbairn: So you never finished your PhD At Stanford?

Bechtolsheim: Correct. I was a little hung up on that initially, but Vinod sold me on the notion that I can always finish my PhD Later, but you can only start this company now, because If we don't do it now, there are going to be five other people that will do this in the next six months. And true enough, after we got funded and got going, there was at least five companies that had exactly same business plan, to build workstations with 32-bit microprocessors, bitmap display, running some version of Unix.

None of these other companies made it. They didn't even get acquired. They all failed. And part of this was, we had the first mover advantage. Any customer we got, they wouldn't get. Any good engineer we hired, they couldn't hire, and so on. So good timing is incredibly important, and of course, the technology has to be ready. At the same time, we couldn't have started the company any earlier either because Berkeley Unix wasn't ready. And we couldn't have started any later, because we would've missed the boat, so the timing was about as good as it could've been.

Kapoor: And you had a target to beat, which was the VAX, right?

Bechtolsheim: Well, the VAX was the standard minicomputer used in engineering at the time, but it was so expensive that even Stanford had only one machine for the whole computer science department. There were 40 people trying to share one computer with one million instructions per second. So in the average you got 0.025 MIPS per person. You could not do any serious computing work with that.

The time sharing kind of worked for word processing and sending emails, but in the engineering world, you tend to run code that actually is compute bound. If you want to route a PC board, you have to map the data structures. You have to look at how to minimize the path lengths, optimize it as much as you can. You need significant compute time and memory to solve these problems. When you get to chips, it's an even bigger problem.

At Sun early on, we actually didn't get a lot of these CAD customers because, for one, the software wasn't ready, but the machines were actually still too small. Our big break came in the '87 timeframe -- when we had the SPARC chip, which we'll talk more in detail in a second, but it was a much faster CPU than the Motorola chip at the time. And to run chip design software, from a company that today is called Cadence, at the time it was called--

Kapoor: SDA.

Bechtolsheim: SDA. Thank you. Suddenly everybody signed up because it was the fastest, cheapest box to run the software. By 1990, Sun had 70% market share in the workstation market, just because it had the right product at the right time. But it did take us quite a few years to get to that level.

Of course, over the next ten years the PC with Windows 32 became fast enough to take over that market. So it lasted for about 10 years, but during the 1990s, the Sun workstation environment with all the tools that it had at the time, was really unbeatable.

Fairbairn: So when you started Sun, what was your own personal vision of what this was going to grow into it?

Bechtolsheim: Yes I was convinced we would build the best workstations and that there was a large market opportunity for such products. One funny story at the beginning of Sun when we raised the initial Series A round, I actually invested my life savings into the company, about $100,000 at the time. Bob Ackman, one of our first two VCs, told me, look Andy, there is a lot of risk here, you could end up losing all your money. I look him straight in the eye and said, I see zero risk here. And I truly believed it.

Because the analysis was, the only company that could actually compete with us, was Digital Equipment, because they were the market leader in technical mini-computing. However they had 100,000 employees and this high price point, and the only way they could adjust their business model was to lay off half the company and they wouldn't do that. They continued to grow their business in minicomputers, but they did not bring a workstation to market until after Sun was doing more than $1B in revenue, and they never achieved meaningful market share in workstations.

Fairbairn: So you never saw Apollo as being a real competitor?

Bechtolsheim: Well, Apollo was our main competitor, but they didn't appeal to most software developers since they developed their own proprietary operating system and were not using Berkeley UNIX.

We had all the developers, so we knew that people would write more software for Sun than for Apollo. Basically, they did some OEM deals that worked for a while, but in the end, we passed them in revenue within two or three years. The problem they had was they started too early. They picked a proprietary operating system, a proprietary network interface, this funny dual CPU structure, and everything was proprietary, it was not an open system. We positioned as the open company and they were the closed company, and that was the end of the discussion.

Fairbairn: They were actually too early. I mean, that was the argument. People wonder why Xerox didn't do more with the Alto, and the problem was that the earliest possible point was the one that you hit.

Bechtolsheim: Yeah, but another thing with the Xerox Star computer, which was Xerox’s attempt to build a commercial version of the Alto, was that targeted office productivity.

Fairbairn: Right

Bechtolsheim: And at $15,000, that was a little pricey for office productivity, whereas for the engineer, it was a no-brainer. So we actually had the right price point for the engineering market, whereas not too many people thought that $15,000 at the time, which is more like $50,000 today, was a reasonable investment for an office worker. I think Xerox gave some Alto units to the White House.

Fairbairn: Well, yeah, the White House actually had Altos running.

Bechtolsheim: Oh yeah, that's true. And they had a laser printer. They were really advanced at that time.

Fairbairn: But you have $10,000 or $20,000 product whereas--

Bechtolsheim: It wasn't the same price point as the Alto. But again, the target was engineers, which had no problem with the price point. And it was an open system, so they could do whatever they wanted to do. Whereas the Xerox PARC, the Star computer was a turn-key office automation closed system. It was beautiful software, they had the Mesa programming language. And they had a nice GUI. It was really well thought-out, it was just the wrong price point. If it had cost $5,000, it would have sold really well.

Fairbairn: OK. So, talk about the transition from the 68000 to the SPARC and how long-- what was the driving force and what was the timeframe?

Bechtolsheim: Back in 1982, the 68000 was the only choice. The Intel 286 at the time was 16-bit only, so they could not even address a reasonable amount of memory or a megapixel display. And it took Intel forever to come up with a 32-bit architecture. So that was actually good for us, because the PC industry, stuck on Intel, couldn't address these applications that needed 32-bit.

But it was also clear that Motorola wasn't moving very quickly because they only had one design team. They just didn't have the resources to move more quickly. They took two years to fix the missing memory instruction and go from 10 to 12 Megahertz clock rate. They moved too slowly to improve performance. The were not on the right curve of improvement.

At the time, RISC got a lot of attention, because IBM has written some white papers about a machine that could execute one instruction per clock, whereas the Motorola CPU took four clocks of instruction, meaning it was literally running at one quarter of that speed. There was work going on at Stanford around MIPS, with the goal of creating a new architecture that would execute one instruction per clock.

Bill Joy, who was really our chief software guru, got all excited about this idea of going four times faster. And RISC was so simple, we could build it as a gate array. And sure enough, Anant Agrawal figured out how to make the SPARC instruction unit, excluding floating point and memory management, fit into a 20,000-gate gate array chip.

This was the first SPARC CPU. Running at 15 megahertz, one clock per instruction, it was four times faster than the Motorola CPU. At the time there was a concern whether we could convert people from Motorola to SPARC, because you had to port the software, but people wanted faster machines so after a few more years all new product development at Sun was based on SPARC.

The biggest problem we had was that the first SPARC machines were servers, which sold in the 100s of machines, and we needed a volume product to drive adoption. So in 1987, I started a project to build a SPARC-based workstation that became our first volume product after it launched in 1989.

Fairbairn: So did you do the SPARC design?

Bechtolsheim: Yes we went to LSI Logic to get a low-cost SPARC CPU designed, and there was an external floating point unit and cache subsystem. The project became known as the SPARCstation 1, which was a cute, little, pizza box product with purple feet. It got launched in '89. It took us almost two years to get it out, but we had to do five new chips which was a lot of work. It was my favorite project at Sun, because it quickly became our fastest selling product.

Fairbairn: So several years before a 68040 product.

Bechtolsheim: Well, at that point, Motorola was shipping the 68020, working on the 68030 which was no faster and the 68040, which I don't think ever showed up. So their problem was they did not keep up. And Motorola itself proposed a RISC architecture. But we already had SPARC, so we weren't interested in that. And the MIPS people want to build their own systems, which was a conflict for us.

Within a year of the SPARCstation launch, the majority of Sun's business was based on SPARC, and the other products just disappeared. At that point the decision was, we'll just build SPARC computers.

And the SPARC chips, of course, improved quickly. There was SuperSPARC, followed by UltraSPARC, which was a much better design, and all the ones that followed that. It became a major design activity.

Fairbairn: And was that your focus too?

Bechtolsheim: No, I didn't work on the CPU chips. I was doing primarily system design. I worked on the SPARCstation 1, the 5, the 10, the 20. And then, the first version of Ultra SPARC, but at that point, I had been doing workstations for 15 years, and I got bored. And I also had some disagreements with company focus and how the company was organized, which did not work as well as it should have. But let's not dwell on that.

In 1995, I became interested again in networking. Ethernet was stuck at 10 Megabits since 1982, which was no progress for 12 years. There were people working on 100 Megabit Ethernet, but if you can do 100 Megabit, why not do 1 Gigabit Ethernet. One would need to build new chips for this, because there were no commercial Ethernet switch chips in the market.

So in 1995 I started a new company, together with David Cheriton, called Granite Systems, with the goal of building a gigabit Ethernet switch, based on a highly integrated chip-set.

Fairbairn: And was there gigabit Ethernet available by that time?

Bechtolsheim: No, Gigabit Ethernet was not even standardized yet, and nobody was shipping it yet, but it was clear that the moment it showed up in the market, it would be a huge opportunity. In September of 1996, Cisco Systems acquired Granite Systems, and the products Granite developed became known as the CIsco Catalyst 4000 and 4500 Series of Ethernet switches.

Fairbairn: Did you actually not ship a product before the acquisition?

Bechtolsheim: No, Granite was in the middle of design. It took another year to even get the chip out, and another year to ship the product. But within two years after the product started shipping, this turned into a $1.5 billion business for CIsco, and this kept going for the next 10 years or longer, so Granite was one of Cisco’s best acquisitions.

In any event, while it is fun to start companies, I actually like big companies where you have a real impact on the market when and the products you develop affects a lot of people's lives. So I stayed at Cisco for a total of seven years.

In the middle of that, I started another company in the video server space with my friend David Cheriton that Sun acquired in 2004. Sun wanted to go back into the industry standard x86 server space, and I led a team that developed a series of products in that space, including Infiniband switches for HPC.

Then in 2004, together with my friend David Cheriton, I started another networking company called Arista Networks, which was incorporated in 2004, however I didn't join the company full-time until October 2008, because I was still working at Sun. Today I am Arista Network’s chairman and chief development officer.

Kapoor: You also invested in many other companies, including Google?

Bechtolsheim: Yes, I did invest in quite a few startup companies, including Google. However I never had an actual venture firm or fund. I simply invested in companies that had great ideas.

Kapoor: And CAD companies, as well.

Bechtolsheim: Yes I also invested into a number of CAD companies.

Fairbairn: So, the investment in Google is well known, but I wanted to understand a little bit about background there. So what year did you make that investment?

Bechtolsheim: Well, the true story is that I met the founders of Google before the company even started, when they were still PhD students at Stanford. Their idea of using page rank to automatically deliver high quality search results and monetizing them with relevant ads was clearly the best idea I had ever seen. My concern was, if they didn't get going, this great idea may not happen and that would be a pity because the world was lacking a great search engine and it was getting increasingly hard to find useful information on the Internet at the time.

Fairbairn: So, I wanted to ask about that. So the idea itself struck you as being a very good idea?

Bechtolsheim: Put it this way. I was familiar with scientific publishing where what matters is not how many papers you write but how many people cite your paper. So if you applied the same idea to the web, you can automatically build a graph, a structure that says what pages are more important than others.

At the time, most people didn't think automatic search was actually possible, because you would need to actually understand the semantic content of each website. Alta Vista, which was very popular at the time, just looked at keywords. But what people would do is they would just add the whole dictionary as a dark page behind the document, so every word you were looking for was in this dictionary, and you couldn't find anything anymore because suddenly every web site thad every word in the hidden dictionary.

Fairbairn: Yahoo was out there also?

Bechtolsheim: Yes. Yahoo believed that you needed human editors, like newspaper editors, to sort out the Internet and that is what they did on their home page. There would be the news section, a business section, a sports section, the garden section, and so on just like a newspaper. The editors would select content from the web that they would present to the Yahoo community on the front page. But clearly, that was not going to scale with millions of web pages, so they just couldn't keep it up.

Larry and Sergey believed very strongly that search had to be automatic, and they had a way to do it. The first demo which they had on a laptop was actually quite compelling.

I got so excited about this that I wrote them a $100,000 check made out to Google Inc — they had the name of the company-- even though the company didn't exist yet, which was kind of funny. I figured, writing them a check would help them to get going.

But I truly can't claim any credit for what they've done. All the credit for the success of Google is due to the founders and the team they pulled together.

Fairbairn: So there were two things. A, you thought it was a good idea, and B, you thought it was very important, because you thought that the search algorithms or the ranking algorithms to the extent they existed--

Bechtolsheim: Yes, I couldn't find what I was looking for anymore on Alta Vista, so I was desperate for a better search tool for the Internet. A lot of my time was spent looking for information on-line. If you can’t find anything on the web, the web wouldn't be very useful. Like, how do you find stuff? That was perhaps the single most important thing.

So for me, I had a personal goal to have something that would make Internet search work well. But, for any new company, the main question is what is your business model? How will the company pay for its investment and make money? I loved this model of sponsored links that would allow an unlimited amount

of ad inventory, where ads could be placed based on the search term the user would enter, which seemed like a fundamentally better way of advertising than banner ads.

I had no idea how this would scale, and I don’t think anyone understood it either in the early days. Relevant ads became one of the most cost-effective ways of advertising. They took full advantage of the fact that the internet is a full duplex communication path, instead of banner ads that were more like television or print advertising and not relevant to the individual user.

Google was an absolute brilliant idea, and their business really took off after the dot com implosion in 2001. I believe what happened at that point that people realized that money spent on banner ads was largely wasted. Relevant ads had the best results, and Google had the highest search market share, so if you wanted your ad to be seen you had to advertise on Google. And once people started bidding for keywords, the price per click went up, but it still provides a very, very good value for advertisers.

Kapoor: So what's your view on the dot-com implosion.

Bechtolsheim: Well, there were a lot of business models at the time that didn't work out. It was a little bit like Hollywood where a lot of movies get funded, but not every movie is a good movie and makes money.

Basically, too many things got funded. There was this incredible increase in venture capital based on the idea that everything would move on-line, but many ideas that received funding just were not very good. The fundamental problem was there was just too much money chasing too few good ideas. So many of those investments that were made just didn't pan out.

Fairbairn: OK. What were some of the other companies you invested in? Google is obviously well known. What were other areas in particular that you struck it rich?

Bechtolsheim: Well, I invested primarily in my own companies. After Sun, I funded all the companies that I started myself or together with David Cheriton. That is a very different experience than writing someone else a check, because you actually work in the company that you put your money in.

On the venture side, I invested,in over a hundred companies in the last 25 years. Many went public or got acquired, so it was a very good return. However I'm doing this less now, because I don't have the time.

Also, there is more money chasing startups these days and valuations are much higher which makes it more difficult to get a good return. Some of the best investments are made during the down years, when nobody else wants to invest, and when only the best ideas get funded.

Fairbairn: So, so it sounds like you've had significant successes. Is there one that you were particularly passionate about that just didn't pan out all?

Bechtolsheim: Well, the ones that didn't pan out were typically in a space that I didn't really understand. The lesson learned is that you have to really understand what you're investing in. You should understand the technology, the market, the opportunity as good as the team itself, because it all comes down to the opportunity, the people, the technology, and the market. Investing based on faith is not a good strategy.

Kapoor: You had a special caring for the CAD world.

Bechtolsheim: Yeah, CAD was one of my first things I did in my career, and CAD is different in the sense that you don't need a lot of money. You need 5 or 10 people to write the software, and see if it works. In the late 1990s, the venture capitalists went into all these internet start-ups and basically stopped investing in CAD, while my view was you could invest in CAD with modest funding and sell these startups profitably to a Cadence or a Synopsys. And in many cases that worked out.

The world needs better computer design tools, even today. It's actually a problem that there's not more investment into that space, and the fact that a few companies dominate the market doesn't help either. But one could improve a lot in CAD, for sure.

Fairbairn: So, let's just go back to, when did you start Arista Networks

Bechtolsheim: In 2004.

Fairbairn: And it's still going?

Bechtolsheim: Of course. We went public in June of 2014, and we are now zeroing in on our first billion dollars in revenue per year, up from $100M five years ago. So we did really, really well in terms of growing the business. The company has a market cap of over $5 billion.

Fairbairn: And what it's focus? I'm sorry.

Bechtolsheim: We are building networks for large cloud data centers. When we started we had some friends at Google, who were telling us that the biggest problem they had was actually not the computers-- they can always buy more motherboards-- but the network. Commercially available network equipment did not solve their problem at the scale they had imagined. Google then build their own network gear. Other cloud companies like Microsoft never got to the point of building their own gear, so those are our largest customers today.

We also focused on 10 gigabit Ethernet and above, so when we entered the market in 2009, we actually thought we were late to market, but the timing was really, really good. The 10 GigE market just took off,

Fairbairn: Timing is everything, isn't it?

Bechtolsheim: The 10 GigE standard was invented in 2000. It took 10 years for this to ship any volume, because the price point was all wrong. Having a standard without the right price point is meaningless. More recently there was a transition to 40 Gigabit Ethernet, which is high run rate now. And next year we'll see a transition to 100 gigabit, and eventually to 400 gigabit and maybe a terabit. So the Ethernet thing keeps going and each generation needs new silicon and new systems, so it's a very busy business because you have to keep upgrading and rolling out new switches all the time.

Kapoor: And the largest content is software?

Bechtolsheim: Correct. Our engineering team is over 90% software engineers. We're not designing our own chips, so on the hardware side it's mostly signal integrity and mechanical design,whereas on the software side, to build software that is totally reliable is a major challenge.

You can't just throw more bodies at the problem. You have to architect the software in the way that you can actually scale it. Which comes down to modularity, and an approach where people don't have to intersect all the time. So the architecture we developed accomplishes that and it has been a huge benefit for more rapid development.

Fairbairn: Do you have major competitors in your space?

Kapoor: Well Cisco, of course. They are the dominant vendor in the space. They're still five times bigger than us in the data center, but we're the only company in 15 years that has gained any market share against Cisco in switching. Everybody else is actually losing market share. And the reason is, we have a better product. When people look at the two products side by side, ours is more reliable.

Fairbairn: What do you think the challenges are going forward? What's your next challenge?

Bechtolsheim: Oh, for me?

Fairbairn: Yeah, you personally. What piques your interest?

Bechtolsheim: I probably won't start another company. A company is a 10-year journey at least, if not longer. I like very much what I am doing at Arista, and we have a huge opportunity in front us to expand our market share and grow into routing and so on. It is a great business opportunity.

I do enjoy meeting young people who have good ideas, and sometimes I write them a check, but that comes down to people who really have some inspiring, new ideas of something that is really different. There are many ideas, but the really good ones are far and few in between.

Kapoor: What went wrong at Sun, in your view, and I know you had--

Bechtolsheim: Should I really talk about this? The post-mortem? Well, a couple of things. I think the first one was in the '80s, we grew really, really fast. But then, unfortunately, a decision was made, which in retrospect, was a very bad decision, to merge Berkeley Unix with AT&T's System 5. The idea was, if there's only one UNIX, instead of two versions, the one UNIX would have more market share.

But that never happened. What actually happened was, it took four years, maybe five years, to finish this merger. There were endless meetings with AT&T, which a few years later wasn't even in this business anymore, as they gave up on Unix and the idea of being in the computer business.

And instead of creating a better operating system, most time was spent just trying to merge two things, which was like painting the flag pole, literally speaking. Do we use this version of the lpr command or this version? And there were long meetings of deciding which dash commands will prevail. It was upsetting to the people used to Berkeley Unix, because they didn't want to change, so eventually they added another command set that was the old command set.

But basically, nothing good came out of this merger and it didn't really help the market share either. Sun would've won anyway. So it was just a bad idea. The real problem during this time was there was no new development on the OS itself. For instance, instead of focusing on building a better Window system or supporting multiprocessors, we lost four, maybe more, valuable years, in this vain effort of combining two versions of UNIX. OK. That was a bad idea.

Then there was this notion of running the company with a whole bunch of planets. Here's the hardware planet, the software planet, the chip planet, and a compiler software tool planet. It was actually Eric Schmidt who said, it doesn't make any sense to charge for our compilers. We should just give them away to get more developers.

But the whole planet structure didn't really work, because Sun had a unified business. It wasn't really selling separate software and hardware. But the idea at the time was trying to unbundle.

Like Apple was criticized for being a closed system. Then they licensed SuperMac-- or what was it called? UltraMac-- to build clones, and the first thing that Steve Jobs did when he came back to Apple, was he killed the clones, because if you cannot build a better system yourself, you don't need the clones for sure. Whereas Sun had this belief that it would help us if there's more people building SPARC workstations, which actually wasn't true.

Then, on the SPARC side, we had an architectural lead over Intel -- 64-bit address space, Virtual machine support, big memory, but that lead eroded over time. And, of course, building chips is hard, and Sun historically didn't have this sort of cultural understanding what it really takes to build full custom chips. The first SPARC chips were gate arrays that were easy. Building a gate array is nothing compared to custom chip. And it took years to really come up with an internal organization that had the technical expertise and the tools to build custom chips. Years were lost getting there.

But the biggest problem Sun ran into was that Sun was a system company that built a combination of hardware and operating system stack with Java tools and so on. But the Solaris operating system got devalued with Linux. Linux, which grew up in the '90s, initially was considered a toy, but by 2000, some large customers were starting to adopt it, and IBM endorsed it in a big way. By 2001 LInux on X86 was getting significant momentum and customers switched from Solaris/SPARC to Linux/Intel servers. So there was a fairly steady but accelerating transition away from the Solaris/SPARC environment. And Sun wasn't really prepared for that. It was a completely different business model, selling Intel boxes. And the company never really recovered from that.

In the end, the high-end business running Oracle databases was still healthy. And I think that was one reason Oracle acquired the company, there was a logical way to combine database and the hardware. But competing in the commodity server business against HP, Dell and so on, Sun didn't have the efficient supply chain infrastructure to be competitive there.

So it was a sad ending. And it's hard to say what decision one could've made to make it better. I wasn't at the company for most of this time. When I came back to Sun in 2004, the hope was that Solaris still had enough differentiation to drive value, but part of the problem was that every software Sun did, was made open source, meaning you could not charge for it because the software was actually free. But if you can't charge for it, you can't make money.

So at some point, you have to charge for software to pay the bills. One cannot just give everything away. There is no revenue associated with lots of free downloads. The open source business really works best for Red Hat and as a support model, where they have millions of subscribers. But it doesn't really work well for a system company that previously sold a combined stack. And that was true even though Sun historically did a lot of open source work including Berkeley Unix, so Sun thinks of itself historically as the first open source company. However it actually couldn't compete with true open source, in a competitive environment where the competition was free software and low margin hardware.

Fairbairn: One of the other areas that Sun was active in, but other competitors grew up was in the network storage and people like NetApp came along. Can you tell me a little but about--

Bechtolsheim: Historical mistakes. Sun invented, of course, the NFS network file system protocol back in 1984. And it was rightfully proud of that. It was the first time you could actually get a file from another computer just by mounting the directory. It was cool. And of course, everybody licensed it, and everybody had NFS. But the notion of building a high quality file server where the file system itself was more resilient than the standard Linux or Unix file system at the time-- that kind of escaped Sun.

Because when you talk about true storage, what you really care what is the integrity of the data and that the data is protected and preserved under any circumstances. NetApp started with that focus and they built a new logging file system. It was just a better design. In fact, one of Sun's key engineers, Steve Kleiman, went to NetApp to lead this development. And quite a few people from Sun went over as well. The moment they had a box, they started selling it, and it turned into a great business.

Fairbairn: And it's just that Sun just didn't think that was important or--

Bechtolsheim: No, it was important, but the problem with Sun was there were so many opportunities at the time. Bill Joy wanted to build a gallium arsenide SPARC chip. OK. So we funded that. Well, that was probably not the best idea, but--

Fairbairn: There are just so many things.

Bechtolsheim: You know what I mean? At one point, Sun was building firewalls. And then said, oh let's not do that, let's just resell Check Point. So then Check Point suddenly becomes this big company. Or we were late building a reliable enterprise back-end stack, so Sun hired VERITAS to do it. And suddenly VERITAS is a big business doing that. Sun put a lot of other people in the business of doing key pieces.

Kapoor: And graphics. I think they wanted to do graphics.

Bechtolsheim: The graphics was another story that did not work out. The bottom line is you have to focus your resources on where you have the opportunity to make the most money, but you also have to execute it well, and there's only so many good people you have, so you can just can't do 100 projects. You can do maybe 10. And Sun, unfortunately, didn't pick the best projects. It was embarrassing to have Network Appliance showing up with a better file server than Sun had. That shouldn't have happened.

Fairbairn: Did that generate a lot of internal meetings and discussions--

Bechtolsheim: No, but it came down to rewriting the whole file system. And then you're like, well, how long does that take? Well, it will take 10 years. Well, after 10 years, NetApp was a big company.

Fairbairn: OK. So related to some of the things you sort of touched on--

Bechtolsheim: Well, many other ideas, like the browser. Sun was actually doing early work in browsers is '93. Then Netscape comes along and, oh, it's just easier to license it. Which was true, but Sun could have been in this browser business much earlier. The first Netscape computers were Sun workstations.

Fairbairn: So, obviously a hot item today is the whole issue of network security and that sort of thing. Is that an area-- a significant element of your current company? Or is that a software overlay issue that doesn't--

Bechtolsheim: Software security affects everything, including switches. Specifically you have to protect the switch against an attack from the outside. So we are making sure that our switches are safe, but we don't get involved into application level security at the networking level, so that tends to be at the firewall level or wrapped into applications or VM environments, that are running protected on the server.

When people break into a company, obviously they want to steal something, typically data. Well, if the data was encrypted at rest that would be a lot harder to steal than having the data in the raw, which is what most people do today. So one of the most important things is to encrypt everything. Which hasn't really happened on the wholesale level. How could people steal 20 million records from the government? Why wasn't this unencrypted? Makes you wonder.

So once encryption of data at rest happens, that should be safer, but then it comes down to managing keys and who has access to what. People just have to get much more deliberate around designing security correctly at all levels, including application level. And there is a shortage of expertise for doing that. Some of the smartest security people work at Google and Microsoft and Amazon, so those kind of environments are actually very secure, so when people sign up with Google, that probably has a higher security level than they have on their own internal data centers.

Computers get attacked every day, every hour, every minute, every second. There are armies of people in various countries that are in the business of trying to get information. And if there's no lock, they will find the door they can open, and get it. So one should assume the country's at war on the industrial side. Why send spies over here to break into the company if you can just reach up inside over the network? Much more convenient. You can’t even trace down who did it.

It's not quite World War III but it is an economic war about getting access to information. It is spying at a scale which was unimaginable in previous years. And once the attacker is inside your company, and they can masquerade as another legal user, they can get and do pretty much everything.

So it's a very challenging situation, and it doesn't matter where you’re a bank and you send money out, or you're designing airplanes, they copy your prints, or whatever it is. Biomedical research. All that stuff that people are investing in, is vulnerable to an attack. It is a major problem, to put it mildly.

Kapoor: So does that impact-- are people asking you to put in hardware or other things in at your level?

Bechtolsheim: They want to make sure that their switches cannot be infected with viruses and do weird stuff, and that's fairly easy to guarantee. But we're not at the application layer where there are hundreds of security companies now that have specialized solutions for analyzing and watching behavior. I mean, there's all kinds of theories how you can find out that you are under attack. Like watch people's behavior and if they do anything unusual, stop them. Because it may not be you anymore.

And it's very embarrassing for the companies. What happened to Sony last year can happen to anyone. The reputational damage, the monetary damages. The amount of stuff that can get lost. It's surreal.

Fairbairn: Yeah, I mean those are quote, unquote, just money or whatever. But then when you look at what is compromised in government databases or military databases that are truly major national threat kind of issues.

Bechtolsheim: And with the Snowden revelations, never trust your administrators. They have access to everything, but who knows what they're really thinking?

Fairbairn: So you came to the US in the 70s--

Bechtolsheim: 1975.

Fairbairn: Do you go back to Germany frequently?

Bechtolsheim: Mostly to visit my family. I'm still focused on high tech, so there's no better place for me than Silicon Valley.

Kapoor: So Sun was a very intense time for you?

Bechtolsheim: Yes, it was incredible intense because it was my first company and my first job actually. We had some challenges early on, and there were lots of late nights and because it is your own company, you have to make it work. It is a very different commitment than being an employee for another company. But it was also an adrenaline rush, because good things were happening every week, and this is what keeps you going.

All good growth companies are like that. There is a lot of opportunity for everybody to contribute, whereas in big companies that are either stable or declining, it's much more challenging. What do you tell people. Like we have to shrink, so hunker down. New companies that displace other companies are a lot of fun, whereas if you're in a company that's being disrupted life is very challenging.

Kapoor: Are you still in contact the Scott [McNealy] and Bill [Joy].

Bechtolsheim: Yeah, I see them occasionally. Bill has retired full-time. He has a sail boat that he enjoys quite a bit. Scott plays golf a lot. And Vinod has his venture capital firm, and he is a very active investor.

Fairbairn: So do you envision yourself taking on any of those activities?

Bechtolsheim: I don't play golf.

Fairbairn: Or sail?

Bechtolsheim: Well, I also have a sailboat and I do enjoy sailing, but I am also fully engaged with Arista. I very much like what I do.

Fairbairn: Not on the horizon. Do you ever, getting back to Germany or Europe, do you get requests or people trying to recruit you to help foster start-ups or other things.

Bechtolsheim: Yes, Germans are coming more and more over here, to figure out or what they can learn from Silicon Valley. What I tell them is that the best idea for German startups is to come here, because in high tech the market opportunity in the US is so much bigger than Germany. For the same effort, you are much more likely to succeed here, and it is much easier to raise capital. You're next to your customers, there are many people who have done start-ups. The cultural understanding how to build new companies and the overall opportunity is just much better here.

Now some companies keep their engineering in Germany, because it's actually cheaper to hire engineers in Germany than here, but you got to be where the market is. If you are not in the US, you miss probably half your market. All of Europe together, is smaller than the US.

Fairbairn: What's the availability of engineering talent in Germany these days?

Bechtolsheim: Well, there are lots of good engineers in Germany. In fact, the number of engineers per capita is higher than the US, including people trained in software and hardware and so on, but they just don't have the same opportunities, so they end up working at Siemens or some traditional company.

Fairbairn: Obviously, the competition for talent here must be ferocious.

Bechtolsheim: Well, that is true. On the other hand, again, you can hire engineers in other place of the world, but you have to be in this country here to sell and market your product.

Fairbairn: Sure. Yeah, you have to be here.

Bechtolsheim: You can build products in other places. A lot of people develop products in India, Romania and other lower cost jurisdictions. You don't have to have the engineers here. and this model of having engineers in Europe while focusing on the business here, I think is a good model for many companies.

Fairbairn: So does your current company have development centers in other parts of the world?

Bechtolsheim: Yes. We have a group in India, and we have one in Vancouver, because it's so hard getting visas for foreigners in the US. It's nearly impossible, whereas in Canada there's no problem getting visas. And Vancouver is the same time zone. It's actually a very nice place. It's just easier to hire foreigners there. We also have a small team in New Hampshire. And we are building an engineering team in Dublin, Ireland. So we were setting up engineering centers in various geographies.

Fairbairn: So is there anything we missed in the Andy story?

Kapoor: Any words of advice for young people?

Bechtolsheim: I do have some advice for young entrepreneurs who are thinking about doing a startup. There is really great information available now on YouTube. One of the best collection of videos is called the Y Combinator Stanford School of innovation, which is like a series of 20 lectures they put together by various people who have started companies.

They teach you everything what is involved with starting a company, how to raise money, how to give the elevator pitch, what it means to start selling, and so on. All the questions you would have if you've never done it before are actually answered in this series of lectures, so these videos are an incredible resource. And of course, you can watch them anywhere in the world as well. Better than any book. You want to hear from people who have done it before, and who can explain what it takes.

Kapoor: So what about the technology areas which you think are most interesting. If you weren't doing networking, or whatever. What area--

Bechtolsheim: Well, the fastest growing stuff right now is still mobile, but it's really hard to be successful there. You are competing with Apple, Samsung, Huawei on the device side, so I would say it's impossible to build a better device, and then there are millions of apps, so to stand out there is a real challenge. But if you succeed, of course, you're big, because it's high volume.

The cloud is interesting, because all these applications that used to be enterprise apps are becoming services in the cloud. It's actually much easier to sell it this way, although you get less revenue upfront. So it's a more challenging revenue model, but long-term, it's much better for companies not to have their own servers, and they can just use the resources in the cloud. Networking, of course, is still interesting.

In Big Data, there's a shortage of data scientists and people who really understand this stuff. So a lot of the work they're doing is consulting people and helping them to structure the data and find meaning in it. But if there was a way to create models more automatically with artificial intelligence that learn from raw data sets, that would be very compelling. Like, here's all this data, tell me what this means.

Kapoor: Does cognitive computing have any future?

Bechtolsheim: Cognitive computing, and by this, I mean the automatic recognition of things has made incredible progress in the last couple of years, to the point where computer voice and image recognition is becoming as good as humans, which is amazing. The computer can operate at the human skill level.

So that obviously has a lot of potential applications across many domains. What it took to compete with the brain was, of course, a very large data center with lots of compute cycles and lots of data. This is the area where we will see the most surprising results.

For example, Google did this project where they trained the computer to play video games, with the only objective function to maximize the score. Within two weeks, the computer was beating any human player, because it figured out anything you can do to get the highest score. While machine learning across many applications is just beginning to be understood, I think that in the next five years, we will see computers do many tasks that previously only humans could do. And that always scares people because what does this means for jobs. Self-driving cars are actually have a harder problem, because it comes down to what you do in an unexpected situation or if there is an accident. Somebody jumps in front of you.

For most decision making, when it is not a death and life situation, you can certainly use computers on a large scale. With the right kind of machine learning the computer learns how to make better and better decisions. That, to me is the most fascinating area ever.

I was fascinated with AI when I came to Stanford in '77, and I took all the AI classes at the time, and even spent a year to trying to build an AI system. But the computers were just too slow and you just couldn't get enough data to do anything interesting. So this only works at the right scale which we finally have today.

END OF INTERVIEW

