--- Start of DINK1.1 Intro (Original).txt ---

Kodak was once almost synonymous with photography and was a major factor in capturing our collective
history, allowing us to share memories with others. Kodak recognised already in the 1970s that
photography could go digital and created the first handheld digital camera in 1975. However,
they initially dropped the product line out of fear of jeopardising their normal film business
and thus delayed their transition to digital. At the same time, they had an 85% market share in
the camera business in the US, maintaining even a 40% share 20 years later. But in 2012,
they had to file for bankruptcy protection, despite their efforts to shift to digital.
In 2007, Nokia supplied 50% of the mobile phones sold worldwide. It was also the year that Apple
launched the first iPhone, with a different take of what a mobile device could be. Five years later,
Nokia's worldwide market share was almost nonexistent, and the mobile phone business was sold
to Microsoft, who later killed the non-successful product line completely. There are many similar
examples, Blackberry, Xerox, Blockbuster to name a few. So is it so that companies cannot transform
themselves or could old and larger companies also innovate? Luckily, there are also examples of just
that. For instance, IBM or car manufacturers like Ford or General Motors.
Stay with us if you want to learn about the difference between innovation and transformation,
which tools help us transform existing businesses or how to make ideas more sustainable
and not just a passing trend.

--- End of DINK1.1 Intro (Original).txt ---

--- Start of DINK1.2.txt ---

When you hear the words digital and innovation, you probably have a good idea of what we
are talking about.
And maybe you also have a specific product in mind, maybe iPhone, maybe Tesla, or maybe
you think of something intangible like a service, a digital platform, or something more abstract
like AI.
But as an engineer, it is important to understand not only the manifestations of digital innovation,

but also its attributes.
What is special about digital innovation from an engineering point of view?
And which opportunities does the field of digital offer those aiming to create the new or
develop the existing?
By a very classical definition, innovation is a new combination of either new or existing
equipment, knowledge, ideas, resources, or other factors.
But does new always equal innovation, not from an engineering point of view?
We should at least demand the introduction of significantly improved products, services
or processes before any digital novelty deserves to be associated with the term innovation.
And since this is about the marriage of innovation and digitalisation, we also need to address
digital.
What parts of digital do we as engineers need to pay attention to?
When you talk about digital, there are some special attributes to be aware of.
Digital products are often easy to distribute, and they are also often easy to replicate
or make exact copies of.
Digital evolution by a basic definition is based on digital signals and the engineering of
devices to use or produce them.
However, any new product or service that harvests this technology is not digital innovation,
even though it seems to be a common misunderstanding.
On the other hand, not all digital development needs to be innovation.
An important part of digital development can be described as transformation, though it
seems to be less famed, it's perhaps the most important part of digital engineering.
Often you are not even developing a product, you are building digital solutions into existing
products, transforming products into services, or supplying a platform for others to build
their products onto.
Sometimes you can take an existing product and transform that into a digital product.
You might take a watch and think about a digital watch as a replacement of that.
And this is more important than you would think.
In fact, great value is often generated by engineers, not by invention or innovation,
but by transforming existing products or services with state-of-the-art digital tools.
Think about transforming the process, doing things in a different way.
Instead of shipping a DVD, you might instead stream a video that you can watch and consume
at home.
So it is important to understand when a certain problem or opportunity calls for a

radical
innovation and when it calls for enhancing or further developing something that already
exists.
Often you need to develop existing products not only because it is the best solution,
but also because it is what the users and thus the market wants.
And being able to profit on innovation or transformation is key to your future existence
as a company, a developer or an R&D division.
So what you need to ask yourself is what do the users want, your career or company just
might depend upon it?
As engineers, we are often driving technology innovations and we are usually really good
at it.
But technology innovation is often what paves the way for new innovations or transformations.
And in order to create value, we need to ask ourselves how does our technology provide
value to the users?
And for that reason, we need to involve users.
We need to center our development and our ideas on users' actual needs and bring them into
the process of innovation and transformation.
User needs are all about creating value for users, but what is value actually?
And how do we go about building it into new solutions?
We cannot talk about innovation without addressing value.
The two things are firmly linked, in engineering so firmly that it is hard to even call new ideas
and products innovation, or for that matter, transformation if they do not deliver value.
But what is value in innovation from an engineering point of view?
A simple equation could be that value is quality divided by the resources that users have
to put in.
And resources can be money, time, and effort, but there is also a societal dimension to value.
Classically, we talk about creating value by innovation and value often has meant creating
an economical value.
In today's society and what we work with, we have to extend that.
So value might also come with social aspects or global sustainability aspects so it does
not necessarily have to be an economical value we create, but it has to be something
that we as a human value and may actually buy into.
Value and value creation, however, is very much a matter of mindset.
What are the mindsets of the users and what are the mindsets of the developers?
Understanding your own mindset and understanding the mindsets of the users is key

to developing
products or services that are not only viable but successful and valuable to different kinds
of users.
Basically, any human mind works in two dimensions when it comes to digital development and these
can be defined in the form of questions.
Will I be able to change and adapt to the digital development?
Will the digital development adapt to me?
Will I be able to affect the digital development in a way so that it suits me?
Depending on how you view these questions, you have a different approach to digital innovation
and transformation.
Based on the two dimensions, we can derive four archetypes of users and developers.
And as an engineer, you need to consider how will my new product or service be received
by the stereotypes?
What that also means is that we have to consider our mindsets.
Where do we start from?
How do we approach problems?
And how do we think about changing ourselves in the process?
So we want to be able to have an interesting and fun life as engineers, also 40 years
from now.

--- End of DINK1.2.txt ---

--- Start of DINK1.3.txt ---

So welcome to this podcast on the landscape of innovation. My name is Per Bekgaard. I'm from DTU Compute and I teach user experience and also work a lot with innovation. And I have an industrial background and have been living through the Nokia ups and downs. I have also my colleague Andreas Bjergsen here. Andreas is also from DTU Compute and is responsible for teaching also computer graphics. And I would also like to welcome Tim McAloon. You are a professor at DTU Construct, but you have been working also a lot with innovation and sustainability in what we are doing. And you have been part of forming our design and innovation education. So the purpose of the podcast today is to unfold a bit of perspective on the landscape of innovation. We may have touched on elements like value creation or the cultures and so on. But Tim, you've been working with building students also that are part of forming the landscape of innovation. So what do you think would be some of the most valuable assets that the students can take with them from our education? Yeah, so our biggest job at DTU is to educate the engineers and the professionals of the future. And for us, the term value is really important for any engineer or any student who is learning how to create products or other solutions that are going to come out onto the market to be received by a customer, a user, whatever type that may be. Value is a very important concept to have the students understand from the very start because value is something which, at the end of the day, is subjective. It's something that you and I may have a different value perception. But what we

generally try to say to the students at the start of their education, at least, is that value is something, if you were to make an equation out of it, now we are at DTU of course, a simple equation could be if we think something has high value, it's the amount of quality we get out over the amount of investment we should put in. And that investment, that could be money, but it could also be time and effort. So from the very first day of their studies, the students that we come into contact with are basically given, you could say, the ammunition to be able to not just judge whether something has a high value, but actually to build in, to design value in, high value into a product or a solution. And let me give a couple of examples of that. One could be that if it's a whole product and it has a competing product or competing products on the market, what is the difference in the value? Why should I buy this product? It should maybe have more features, more functionality for the same price, or maybe the price could be higher, but if the functionality is so much bigger, I'd be willing to pay for that. It could also be down to the very minute feature level of a product, meaning that I could introduce as a designer a feature which is easier to use for the user than the predecessor feature or than a competing product's particular feature has. And that means that the effort that I have to put in as a user is less, which means that my investment is less, therefore I would evaluate something as greater. So although value is, you could say, a subjective term, we try and equip the students to think of how do we not just examine or analyze whether value is there, but actually to design it in. And there's actually a number of design tools and methods you can use to make sure that you're designing high value solutions. Three things you need to think about before developing any tech product or service. So which methods or tools do you think that it's important that the students sort of take with them from their education here? Yeah, well, one for example is to know the field. What is it we're actually developing of a product or a solution? We need to know what the competing solutions are, first of all, and you can do that by doing a teardown or any form of a benchmarking analysis, of course. That's looking at what's the foundation we're standing upon. The second is to understand if I have a great idea, how do I actually crystallize that idea into a solution? And is that solution actually going to be better than the platform or the baseline I've just created by doing the benchmark or the teardown or whatever it could be? To see what better we're going to do based upon what's currently available. The third area is to actually look not just at the technology, but actually at the user. So does the user actually want to use the solution and what is the value add that we're creating here? Because if we end up giving a new solution or a new functionality which is not really designed, and there's many examples of those out in the technological world, which I think UPL would probably be better at giving a long list of than I am, but it's about making sure that we set some goals and some guidelines there. So you could say, to answer the question whether is there a mindset that we can give the students or give young professionals to work with the value term, there are certain ways to go from a technology-led value creation and also a user-driven value creation, which is understanding user needs, understanding the technology availability, understanding the amounts of the limits, the upper limits to where we should push the envelope of possibility depending on the novelty of the product or the solution and so forth. So there are frameworks and ways and processes that we try and convey to the students in a systematic way to think about the value in their product development. So in my view, I probably more come across sort of technology-driven development rather than sort of user-centered or user-focused development. And I

don't know if you agree with me, but I think that as engineers, our biggest risk is being very technology-focused and forgetting a little bit about the users. Is that also how you see it, or do you think that we have kind of moved the mindset during the last couple of years also to focus more on the users? I think both. So way back when we started the Design and Innovation Education Program, which is exactly 20 years ago this year, actually, one of the reasons for doing this was to exactly introduce what we call a socio-technical competency to engineering and to engineering studies because we can be the world's experts in technology development and in designing and constructing technical solutions. But if we don't understand the socio-technical aspects of that, and socio-technical means not just the user, but all different stakeholders in society, and situate ourselves with the solutions that we're creating and building, then we can really make some wrong turns and go down some blind alleys and waste a lot of resources, and at the end of the day, actually not serve the ultimate goal or the ultimate receiver. I think DTU has changed and developed, and the world has changed and developed in many ways over the last 20 years, and if I were to focus on the way in which the education and the way in which DTU's general activities have developed over the years, it's unrecognisable I'd say. And I think that one of the main added ingredients to the education, to the way in which a student and a staff member both operate at DTU is, I think, in a much increased manner with the user in focus. We have a slogan, technology for people, for example, I think that says it all. We have a big focus on innovation and entrepreneurship, which is one of the four legs that we stand on at DTU as staff members. We support the students with technology to create solutions for people, and actually support them to bring solutions to the market as well. So I think there's many things in the way in which we operate, and I think that I haven't seen to every course that we offer at DTU, but I think that when I speak with students who, of course, most of the students I speak with are doing the design innovation programme, but take different specialisations, they come back and they can relate directly what they're doing to their field, which is much more user-driven, I'd say. Digital development, how is it special? So Andreas, now we're talking about innovation, I think, in general, but do you see that what we discuss now also applies to the field of digital transformation, digital innovation, or are things different when they are digital? I mean, digital is better, right? Well, no. I don't think necessarily that digital is better. Digital is, of course, very different.
footprints. I think the only solution for them is to dematerialize their products, to go to services, products as a service, maybe software as a service which you'll know more in the field you're working in. Materials as a service is a term which has been spoken about and all of these things can only be there with digital solutions as enablers. So the way which I see digital technologies and digital solutions is for sure as enablers. Even though I use them as products myself personally, but in a research perspective and a lot of my teaching where I bring elements of digitalization and digital solutions as you could say as elements of how to design and innovate for new value propositions, I see that for sure digital solutions as being enablers. May I just ask a follow-up question? Because there's one thing that concerns me a little bit when it comes to using the digital technology to enable resource sharing, which is that it could lead to the opposite of democratization of the technologies. Because if you share manufacturing capability for instance, then you're almost necessarily concentrating the manufacturing in some gatekeeping organizations and these could be for instance in

China, to take a not-so-random example. And maybe you would like some of that stuff in Europe too. So I don't know, do you have some perspectives on this conflict between democratization and... I have some personal fears and apprehensions, yes. I think that free market forces often make that if a solution is a really good idea, there'll be other competing solutions out there. So I think that if we can halve the amount of capacity, or rather the amount of structural waste as it's called, in our capacity for manufacturing, then we're doing a really great job. And structural waste means basically we have facilities which are there and not being used to their full potential. So this building for example is for many percent of its time, I think 70 to 80 percent of its time, is not being used to teach students or to research. And you could say well that's just the way it is, or we can say that's not acceptable. And if we take the building and say well could we build it to have different purposes? Slightly more difficult for a building, but for manufacturing capacity for example, that has a particular raison d'etre, a particular reason for actually being put into operation. Let's take for example distilling or beer brewing. I'm not sure why I'm thinking about two alcoholic examples there, but just for example, because they exist actually, there are some brewing companies which actually rent out their brewing space to principal competitors. When you go to the supermarket you can buy the one beer or the other next to each other and they're actually competing. But they've actually made an agreement that well it's not really the brewing capacity which necessarily is the competing element. Why don't we collaborate on that? Because then we can save costs in these areas and then from that perspective you could say the competition then becomes the marketing, but also the ingredients of course, and the way in which we bring it forward. So yes there's always potential downsides and potential dangers of focusing manufacturing in one place or focusing capacity in one place and you could say de-skilling ourselves as well for sure. But I think there's other solutions where you could maybe make cooperatives where everybody goes into something with the same interest and the same access you could say. So perhaps related to that, so a model that is often used I think in digital is like that of horizontalization. So you would see sharing of resources like in an Amazon workspace or you know any of those cloud solutions that somebody invests in the infrastructure makes it available to everyone else and that's of course an enabling factor for further innovation I think. And I was thinking one of the other things you mentioned previously was about like Apple or others taking a cut of the cake. So it seems like sometimes innovation at least on digital side is that we have some isolated innovation specific products but it's building on top of a lot of existing services that you grow. So is that also where you see a challenge of sort of somebody taking a control point or what's your thought on that? Actually reminds me of an XKCD cartoon where you have this this digital product and it rests on a lot of a stack of technologies right and at the bottom there's this tiny little open source project that somebody maintains for free somewhere. So I think that maybe the concern is almost more brittleness that we can have some sophisticated products that ultimately rely on some not very well maintained packages or maybe packages that are well maintained but maintained like by one dude and if he or she stops maintaining a lot of people are in trouble. So there's one example that pops up sometimes which is the GUI tool called DearMGUI that's enormously used and maintained just by a single person. So I think that's maybe my main concern the brittleness that this might introduce. The burden of legacy. Can I just ask a question to this because I think it was nice to hear the devil's advocate view on what could go wrong and I think we for sure need to be

careful. I think that many companies that I'm experiencing through research and other collaborations are trying to make this transformation exactly to start to decouple their value creation from physical products because one it's better environmentally but two it actually is easier to scale. What is your experiences in terms of companies when they do try to start this transformation to more digital solutions? Is it easy for the legacy company to build in new software solutions to develop their software to develop their organization in fact and the colleagues they have around them to be able to bring more digital solutions into their work? I would say as a young engineer and I was looking at stuff that was out there and I thought okay we can do this better so you jump into rewriting it and it takes not only a day or two but it takes significantly longer until you sort of have re-established the functionality that was there already. And for me having worked long nights and trying to recover stuff I kind of learned that there isn't the hard way and that if you have a legacy and you have a legacy burden it's a significant element that you simply have to take account of. So I think your point is super relevant that if you transform a business it's not just about creating a minimum viable product something that just pops up and delivers a nice new service that nobody has heard about. It's also about fulfilling the expectations from existing customers and I think there's been quite some it's a drastic changes in the in the landscape also I mean if you go back 50 years you might have heard about other companies maybe 40 years you have digital equipment corporation was one or Apollo or others that were delivering computers and they were you know the large players at some point they're gone. So transforming your business and understanding these disruptions that can allow you to do things in a different way that's super important and it's not so easy. And I think some of the challenges perhaps is also down to the culture. The way that we sometimes as individuals resist kind of change because we see that we're going to lose something and it's not clear what we're gaining in against that. So it needs to have sort of an active mindset of doing that and there are companies that are capable of doing it out there but it's it's something you have to be very much aware of. What are the biggest barriers for digital innovation and transformation? I did a case study together with my research group a few years ago on a British company which is making it's actually MAN truck and bus UK so it's a German owned holding company but it's the British branch of that and the former CEO in the case study which is published that's why I can talk about it he told that when they were transforming from a straight product business to a combined products and service business which involved lots of sensors in the trucks a dashboard not a traditional truck dashboard but a dashboard for the truck owners the haulage company to be able to see how well the drivers were performing and so forth. All that was relatively easy to build in because it was something new but the transition of the business from just selling trucks and buses to actually selling performance on top of.
He said it took him 10 years. And there are all sorts of barriers along the way, which you also just mentioned, the culture and the way in which we sell the solutions and so forth. From your experience in the industry and also observing the industry since you've been at DTU as well, what are the biggest barriers or biggest hurdles to watch out for? I think definitely culture is one of the biggest barriers and sort of having a mindset, being open for transforming your business and doing things in different ways. I think often an organization tends to reproduce itself and wants to have this built in, staying alive urgency. But I also think that understanding your customers and where the business is moving, it's super

important. I mean, if you were selling horseshoes 250 years ago, you could probably sell quite many of those. They are sort of slightly less used today because the transportation industry has transformed completely. So understanding, I think also what your core business is and how you create value for your users is in my view also a very important element and perhaps even more so creating a shared understanding of that. So you talked about the 10-year transition period. Nielsen Norman Group have made some analysis of the sort of the readiness of companies to transform their business into more sort of a user-centric view. And I think they are talking about 30 to 40 years horizon from an organization where deep down the customer is seen as a hostile thing that we don't want to talk to until the customer actually becomes central for what you're doing. So having a shared understanding of how you create value I think is also an important element. And if that is lacking, then I think it's going to be very difficult to do a transformation process. What about luck? I think we should also talk about luck, to be honest. So an example came to mind, right? We are all of us carrying several ARM processors and why are we carrying them? I guess in a way it's your old company we to some extent have to thank, right? But why were the ARM processors available to them? And I think the reason was really that, what were they called? Acorn or something. They really wanted to make computers, but they couldn't because the market was kind of sewn up. So what do they do? Well, perhaps we could license our platform. And then they started trying to find licensees. And I think actually the first one was Apple and their Newton. Now nobody bought the Newton, which was a problem, but along came Nokia, and they pushed all of the ARM chips into the Nokias and then later into the iPhones. The rest is history. But if they had been able to make computers with their chips, maybe they hadn't come up with a licensing model. So yeah, I don't know if the listeners really gain much from this, but it seems to me that serendipity is a huge part of being successful in a digital transformation. Actually, I think there are quite many examples of this kind of luck. Or maybe sometimes it's just the fact that you run out of money, and then you have something, you're trying to put it in the market. I think Flickr or some of those early photo sharing services actually came about in that way. And it's back to what we maybe touched on previously, that if you cannot verticalize your market and get into that, then maybe you try to horizontalize and then make sure you can compete with everybody else by making your platform available. So I think probably what ARM have done is also to share kind of quite openly also their plans and what they want to do and have been open about that. And I think they've built a trust, so it's sort of an organizational element to it also. But yes, I agree that it's probably a bit of luck that make that happen, I think, at the beginning of the day. In entrepreneurship speak, they call it pivoting. Pivoting, nice. And I think pivoting is all about turning serendipity into something systematic and saying how can we make sure that we are able and flexible and agile enough to be able to pivot if we see something coming along. And I think that there's many, many examples, especially of startups, which it is serendipity and luck. And I think that the idea that with training and with coaching, we can see that there are actually a number of different routes that we can take from the verticals or the horizontal, as you mentioned, Pierre, and other ways of actually making sure that we make our own luck as well. We have all the options, not all the options, but some of the options covered, and we have a contingency plan and so forth. And that's one way of harnessing serendipity. The societal aspect of digital innovation and transformation. So, Tim, we talked a lot about value creation also, and value is,

of course, not just something that is monetary value. Maybe there are other ways of ascribing value to things. So to wrap up, could you sort of perspectivize that a little bit for us? What is value creation also in this landscape of innovation? So we started actually talking about value creation. I was talking about the definition of value as being a subjective thing. If we talk about how to evaluate whether we've done the right thing or we've done something which is creating value, and if I were to use the framework which I'm very comfortable with using, which is sustainability, many companies talk about the triple bottom line. And this triple bottom line is both environmentally sustainable, economically sustainable and socially sustainable. And I think that if something is to be of value, nowadays we cannot get around all three of them being evaluated. And actually when we teach our students in innovation classes and within product development classes, we also teach them today, there's no way around it. So environmental value or environmental improvement means that a solution should be better than its predecessor, at least a little bit better, if not much, much better. And we can aim for much better. I think digital solutions can really enable us to do much, much better. If we talk about the economic sustainability, if you're going to be in a business, that's of course what they're going to go for is economic sustainability. And I think it's not just about the money, but it's about new ways of doing business, new ways of actually operating business. And I think that digital solutions can help us to do this pivoting, to do this changing in the way in which we actually earn a business, which is not just about pushing products out of a factory, but actually pushing value out and making sure that we have value creation. And then the third one is social sustainability. And I think there's very few areas which are like the digital sphere in terms of all of the potential social barriers or social hiccups or social minefields that are out there about we can do things with all the best intentions, but if we don't take care, we can be hacked, we can do the wrong thing, we can share the wrong data and so forth. So there's something really, really important there, which needs to be considered is all the ethical and social sustainability of our solutions. So again, I tend to use the sustainability trouble bottom line as being a way of evaluating any innovation today. And I think we, to an absolute high degree, can do it in digital solutions. So for our students, the digital innovation is kind of a possibility of going out and creating value in a new way, perhaps fulfilling a lifelong dream of creating value for humanity. So thank you very much, Tim and Andreas, for being here today. It's been a great pleasure and hearing your experiences and being able to tap into all of your wisdom. So thank you very much. Thanks for the opportunity. Pleasure to be here.

--- End of DINK1.3.txt ---

--- Start of DINK2.1 introvideo_final (Original).txt ---

Welcome to Digital Innovation and Scalability, the second module of the Digital Innovation Canon, a collection of online teaching materials made for DTU. In this module, we will go in depth with scalability of digital solutions from an engineering point of view. Which questions do we as engineers need to ask ourselves? Not just when we are designing solutions meant for rapid scaling, but also when we're designing solutions not meant for large scale. Is there in fact both risks and opportunities to our product or our business hidden in both

approaches? We will dive into subjects such as virtual versus physical resources, value creation through users, product versus service, scalability, cloud solutions, advantages and disadvantages, green computing, complexity and security, attack surface, and user management. To present the latest research within these topics, we introduce a series of researchers appearing in videos on the podcast in the next four chapters. And for a real world perspective, we introduce video interviews with three highly successful Danish companies, all based solely on digital innovation, MobilePay, VU, and Corti.

--- End of DINK2.1 introvideo_final (Original).txt ---

--- Start of DINK2.2 per_-_clarification_of_concepts_final (Original).txt ---

In this module we are going to unfold different aspects of scalability. I will first take you through some of the key concepts and some of the considerations that are important, and based on that and some questions that we will ask, you will be able to explore more in depth through the rest of this module. Scalability is the ability of a system to suit your different needs over time. It might be growing with the number of users, but in some cases it can also be shrinking as you have periods when there are less users that are using your systems. You can do this in different ways. You can scale systems up by adding more computers, or you can do it by also buying faster and more expensive, usually, computers to deal with it. But it is also sometimes possible to design for scalability. Maybe you don't need to store all of your data in one large database. Maybe you can decentralize. Maybe people that live in Los Angeles don't know about what is being sold here in Copenhagen or the other way around. That is some of the topics that we are going to deal with today, and we will also be digging into, for instance, some of the aspects of security and how you can design your systems to be able to respond to your users' needs. Scalability comes with some risks, but also some opportunities. In terms of the risks, then availability, performance and reliability are issues that you need to consider. So, are security issues, the increasing amount of users might cause more attack points, and you also need to have a way to safely manage your users and the roles they have in your systems. When we use a service, we are sometimes okay if there are minor glitches or if there is a functionality that is a little bit odd to use, as long as we actually find the service overall useful. But one of the things that we are not really prepared to deal with is lack of availability or lack of reliability. There are some good recent research papers also that shows that we are willing to deal with certain issues, but if we remove functionality or if generally there is a lack of availability, then we actually fly away from those systems. We don't even want to fight them, we just uninstall the app and we leave the system. So when you are designing systems, it is important to consider how you make them available so that you can access them, how you make them perform so you don't have to make people wait a long time to get a reply, and how you make it reliable so you get consistent and desired answers to your prompts or whatever you are engaging with. Sometimes these are solvable by scaling up the systems, but sometimes they also need design considerations, how do you divide the different parts of your system, where do you store data, is it stored locally on your computer or your access device, or is it stored somewhere in the cloud. And that might be issues that you have to consider already from day one when you design

the system. Let's also look at the opportunities. So sometimes a growing amount of users actually create more value. If you are the only person in a car sharing system, you don't actually have that much benefit from it. And you could even say that even if there is a good number of users, if there is nobody else that are near you or share the same transportation patterns, then there is actually not much value in such a system for you. So that refers to what is called the network effect, that more users can create more value. The network effect has sort of different phases and different ways of looking at this. So one model talks about a cold start problem, and then the next phase is the critical mass, and then you might go to a bandwagon or an inflection point where a lot of users can join. So let's take those for a bit by bit. So in terms of the cold start problem, that deals with having enough data or having enough user to have at least some value for those that are joining your service. I already mentioned the example of the car sharing, but there could also be other places where you want to have enough data so you can offer, for instance, a set of recommendations for users. Netflix started out this way by shipping DVDs so that they could get some insights into what people would be using, and they used that data to solve part of their cold start problem. At some point when you have enough users, you get through this critical mass, so every new user can find somebody to interact with or benefit from the data that is already in the system. And if you can keep attracting users and your systems are scaling well, then you will probably see an inflection point where you can just increase the number of users, at least up to a certain point where every new user adds value, or at least as long as you can keep your systems available and performing and still offering the reliability that you wish for. There could also be other benefits. You might be able to attract third-party users or third-party collaborators. Perhaps you can offer services that you would not be able to create on your own, but you can do that because you have enough critical mass. There are many examples of this. Perhaps Amazon is one quite well known, where a lot of other companies or smaller companies are sort of collaborating. They bring some added value and they might sort of join that ecosystem. And it might also be that your transaction costs per user will go down. Perhaps you need to have some expensive equipment or some expensive services that you have to have running all the time, but sometimes if you scale your systems well, you can actually lower the transaction cost per user or per transaction through your systems. So scalability is about growing your systems, or perhaps in some cases shrinking them, so that the costs of running the systems actually match your users' expectations and you're able to deliver the user experience to the users at the end of the day that they expect from your systems. We will unfold that through a number of chapters. There will be a few short mini-lectures that helps you to understand some of the key concepts. There are some really, really interesting case studies that we will delve into. And there's also a podcast that tries to unfold some of the dilemmas that we are in reality facing when designing systems.

--- End of DINK2.2 per_-_clarification_of_concepts_final (Original).txt ---

--- Start of DINK2.3 case__mobilepay_final (Original).txt ---

So, mobile pay became an idea in 2012. There was a lot of movements in the banking sector and the payment sector to launch mobile payment solutions and especially in

Denmark there's been a long tradition for banks working together on various digital infrastructure, payment infrastructure, the Dancard, our direct debit solution, Betaling Service and other solutions, NEMID. But Danske Bank, based on experiences with mobile banking, decided to go its own way. So there was basically a race to get first into the market, Danske Bank against all the other banks and Danske Bank launched five weeks before the other banks. And why was there this interest in this race? There was a big interest from the executive management in Danske Bank to launch new innovative digital solutions that could also scale across the Nordic countries where Danske Bank is present. So it was an opportunity that was seen in the market and a competitive dynamic between Danske Bank and the other banks that made this come about. So we started to focus on where can we solve that pain point of not having cash anymore and therefore we decided to launch with the person-to-person or P2P transfer as the first solution, build a user base from that and from then on expand with other solutions. We focused an extreme amount of time and energy on simplicity because we realised that to win in this market you need to have the most simple solution. So we looked into how can we cut away all the unnecessary steps in the process of a mobile payment, for instance when I transfer money to somebody, and cut it down to five simple steps. It takes seven seconds from you open your app to you're done and out again. So we won by making it extremely simple, by being first, but also by making it extremely simple to sign up. If you compare MobilePay to the competing solutions, the competing solutions took forever to sign up. You needed to go to your computer and enter codes and so on. With MobilePay you could just enter your information directly in the app and start to use the app in a few minutes. And then lastly we also, a bit by chance, but by focusing on this person-to-person transfer, we created a viral element, you could say, around the solution. Because if I send money to somebody who's not yet a MobilePay user, they got a text message saying there's some money waiting for you, download the app to get your money. And of course that created an incentive for other people to start using the app, so it spread extremely quickly and was an instant success. 100,000 users the first month, another 100,000 the next month, and so forth. And in a country the size of Denmark, that made us quite quickly the most used solution. In Danske Bank the mantra was it's better to disrupt ourselves than to leave it to somebody else to disrupt us. So there was really a clear strategy from the executive management in Danske Bank to do something that the bank couldn't do itself, and do it sort of on the side of the bank, and also take risks, if you like, take chances, bet on solutions that you couldn't do as a normal traditional bank. It's fair to say that when we started out, we had never ever imagined a growth like the one we then ended up seeing. So both from sort of a risk perspective and a cost perspective, that hadn't really been factored in in light of the size we now have. But we then also did things to contain the risk. So mobile pay is to a very large extent a closed ecosystem. We know who the users are that send money to each other. You cannot send money to somebody outside Denmark or outside Finland. All the merchants you can, all the shops where you can pay with mobile pay are also customers of mobile pay. We know who they are. And luckily in the countries like Denmark and the rest of the Nordic countries, we have fairly good public registries of businesses, for instance. We have strong IDs and digital IDs for consumers. So by doing that, so a combination of strong identities and a closed ecosystem makes it fairly easy for us to lower the risk. And we have an extreme low risk, for instance, compared to, and fraud levels, low fraud levels compared to card payments, for instance. So it's due to the infrastructure, the

setup, and that we didn't, you know, overnight open up for the whole world. But we sort of limited it to Danes living in Denmark, having a Danish account number, having a Danish phone number and a Danish social security number and so on. So on the technical side, we did a number of things. But I would say the majority of the activities behind mobile pay were based on very keen focus on the user experience and not so much on the technology. But how can we design the user experience to be as simple as possible? And in order to do that, to have the very simple solution you have in mobile pay today, where at least as a user, you only see very simple screens, but behind the scenes, there's a lot of stuff going on. So we had then, in order to make that happen and make it so simple, we had to work a lot on the user experience, the user journey, on the legal aspects of being a user. What can we do and not do with the information we have? How can we navigate in being a regulated company with a license from the Danish financial authorities? How can we manage their requirements? And on the technical side, it was not because of the technology itself, maybe the governance around the technology. If mobile pay had been a traditional project within a bank like Danske Bank, managed with all the governance and the traditional success criteria and approvals and so on, we would have launched 12 months later. So the biggest challenge on the technology side was actually to, on the one hand, work within the setup of Danske Bank, and on the other hand, do everything differently. So work much faster, more agile, get approvals much faster than you otherwise would. It was not the coding in itself, it was managing the structural complexity of a project like this. And there, of course, we were helped a lot by the wish from the management in Danske Bank to set us free and let us do things differently. Mobile pay was also, to a certain extent, a response from Danske Bank to the potential threat competition from the big techs, the large Chinese and American companies and other fintechs that are starting up. So in that sense, having a Nordic scale, having a strong Nordic presence was key. It's also, to a certain extent, a market where the winner takes it all, especially on the person-to-person transfers. Nobody wants to have five or ten different apps on their phone. So when I need to transfer money to one person who is the customer of one bank, then I need to use this app and another app and so on, which is the situation we have in many other countries. There you have separate apps per bank. We wanted to be this Swiss army knife, you could say, the one that could solve these needs across all banks. And that required scale, and to scale fast to get there first, to be the preferred app. So has everything then been a smooth ride? No, it hasn't. In the early days, we had this principle of simplicity. That is still core of our values, what we want to do. We say we simplify life. That is our mission in the world, to simplify the life of both consumers and users and the merchants that use our solutions. Over the years, as we've added more and more products, more and more features to our app, we of course sometimes have had to compromise, you could say, on the simplicity, because it's fairly easy to make a person-to-person transfer. It's much more complex to build a solution for online commerce, in-store payments, paying your bills and so on. So when we have failed, it has almost all the time been when we have not remembered our roots about simplicity. So in order to get something out there in the market fast enough, we have made something that wasn't simple enough to use, where we haven't focused enough on what is actually the pain point that we're going to solve here. A pain point for you as a user, or a pain point for a merchant that has to accept payments, whether as a bill or in the store or online. Today, we are 380 people in MobilePay. After the merger now with Vips, we will be 700 people. It's still a mix

of people who have, you could say, traditional banking and payments skills, but also people who consider themselves to work in a fintech environment, and creative people, but also legal people. We have, you could say, all kinds of compliance, risk, customer support and so on.

--- End of DINK2.3 case__mobilepay_final (Original).txt ---

--- Start of DINK2.4 veo_case_final (Original).txt ---

At Veo we make an amateur sports camera, mainly for football, and the product is a camera you mount it next to a football field, then you take your phone, open the app, press record, then it records the whole match, and it makes analysis of what's happening, it detects the goals, it makes a follow cam following the game, and makes a lot of analytics that the coach can use afterwards. The Veo camera can also live stream the football matches, so people can, family and friends can sit at home and watch the players play the game live while it's happening. This idea of filming every football game, we tried with GoPro cameras and all sorts of telephones and all that, but it didn't work out the way it should, so we had to make our own physical camera, and it just took off that way. We made a camera that could see the whole view of the football field, and from that on we made a follow cam that could follow the game where the people and the ball is, and now we have our own internal AI that works on being able to follow each player and give value that way around. So our product is heavily based on a lot of engineering, so there are a lot of engineering hours going into the product, which means that we also need to have a lot of customers before it becomes a good business. I think from the beginning it has been, it's a long term, there's a long term plan to get a lot of customers before it becomes a viable business. Our users are actually our customers, they are the ones who pay us, so our business model is different from, for example, Google, where the users are the product, and of course the more customers we get, the more valuable the product will be, because the customers can help each other and can share content and can share ideas for how to use the product, so there are online communities where people talk about the product and help each other to use it the best way. Our main source of income at Veo is subscription. People pay every month for using our product, and that's mainly because we have a lot of software and a lot of processing going on, the video storage and processing costs money in the cloud, so that's what the customers pay for. Regarding security and keeping our business secrets secret, we have had the strategy, generally, not to try to hide things too much. Our strategy has more been to develop fast, so we'll be ahead, because even if people steal our ideas, they'll probably be half a year behind us anyway, so we will still be ahead. That's been our main strategy. But when we launched our VeoCam 2, so the VeoCam 1 could do the recording and the processing in the cloud, the next camera, VeoCam 2, could do the live streaming. One of the reasons we can develop fast and make new products is that, contrary to the medical industry, for example, our models don't have to be 100% precise. If we make a wrong processing of a video, it will not be a disaster. It's okay. It's most important that if the majority of the videos get improved, it's okay that a small percentage are bad, and this means that there is a lot of room for our AI group to develop new and creative models very fast, so it's a very interesting field to be an AI developer in. So our processing of the videos happens both in the cloud and on the

camera. So our processing of the videos happens both in the cloud and on the camera. There are pros and cons for doing things on the camera and in the cloud. If we do it on the camera, then we can do the processing right away while recording, so customers will get the result faster. But that also means that we need to put more advanced hardware in the camera, so the camera will be more expensive. Scaling up means that we had to change the way we do our software development a little bit, because when it was very small, we could do rapid development and push things out, and if something was a little bit off, we could tweak it and fix it. But as the company grew and we got more customers and more edge cases for our software, we would spend more and more time on fixing the edge cases. So this means that we had to, while as we grew, we had to slow down a little bit how fast we developed, because if we continued to develop at a very high pace, we would get so many edge cases that our development would actually stop, because we would have to fix edge cases all the time. So our cloud engineering team has worked a lot on automating everything, so we got rid of all these manual edge cases, and that has been a necessary process for doing the scale-up. So our product is both software and hardware, and having this physical product means that some of our development takes quite some time. When we start developing a new camera, it takes typically two years before it's launched on the market. Yeah, it's a challenge with the hardware, that we have so many good ideas that we want to sort of publish right away, but it just takes more time to develop the hardware. VU Headquarters is comprised of both production on the ground floor. We also have camera development on the first floor, and on the second floor we have a sort of back office. So we have all people, all kinds of different people working here at VU. It makes a good environment that everyone is involved in more or less everything. We try to make sure that everyone is involved in everything that we do. It makes a good environment that everyone is involved in more or less everything. We try to not work in silos, but work very cross-functional. There's a lot of communication between production and development, and whenever there are issues with the production, they come up to the development department, and things are fixed very rapidly. We had, for example, an issue last year where the button on the VU Cam started breaking off, and they noticed that, got that information from customer support. Production found out why it happened, and then came to development and asked, what can we do about this? And from the issue was detected till the mechanical engineers had a 3D-printed solution that was in mass production, it was less than 24 hours. And that would not have been possible if we had our production in China or Vietnam or Mexico. VU is focusing quite a lot on our culture here. We call it culture first. That means that the whole building here is quite unique. We have a sort of a start-up feel. The community here is quite good. This is definitely a good place to work. We have so much fun, and having fun is a big part of working at VU. We are still quite busy, but we do try to have fun every day.

--- End of DINK2.4 veo_case_final (Original).txt ---

--- Start of DINK2.5 corti_case_final (Original).txt ---

The idea behind Cordia was that we founded the company from the vision that we needed to do something with healthcare. We believe that we solve big problems in the current way that healthcare is working. We saw that in a country like Denmark,

we saw it in the rest of the Scandinavian countries. And when we looked over the ocean all the way to the U.S., we saw even bigger problems. We believe that if we could build a technology that could help optimize the quality towards the patient in healthcare and also alleviate some of the biggest stress points that they're working with right now, then we would have a good and well-functioning company. The big problem that we set out to solve in healthcare was in the patient interaction. So there are a plethora of different patient interactions in healthcare where patients are calling into, for instance, the 112 line or calling into the hospital, calling their general practitioner, having a video consultation and so on, where either these interactions are not sufficient, so the information that is disseminated in these interactions is not enough to make a good healthcare decision for the patient, or they're simply too time-consuming. So the first problem that we had our stellar focus on was the 112 conversation. So when someone is calling into a 112 line in Scandinavia or a 911 line in the U.S., they're in dire need of help. And most people that are calling that number is in dire need of immediate help and critical, acute help. And they're expecting an ambulance. But for someone to send an ambulance, they need to understand what is wrong and why they should send that ambulance. And that conversation itself is really difficult to understand for the healthcare professional. And we believe that if we could build a technology that could help out with that conversation, make a better decision based on what is being said, then we would have a good solution to some of the problems that healthcare is having. The problems that we wanted to solve in healthcare is a very broad problem. And it's difficult to make tangible. I come from machine learning and artificial intelligence as background within that. And we believe that if we could build AI that could understand the dialogue between the patient and the healthcare professional in order to structure the information given in that conversation, then we had a good technology that could help make better decisions. So what we set out was basically build an AI system that could understand the dialogue similar to the dialogue that the two of us are having right now, and take all of that information in that dialogue and take out the salient part, so the most important part of that information, structure that information, and show it to the healthcare professional for better decision making. So that ended up being a software interface for the healthcare professional. While they had the dialogue with the patient, they were looking at our software interface. And when the AI was prompting something, for instance, did you ask about breathing patterns, did you ask about whether the patient is conscious, etc., then the healthcare professional would get those cues, and they could start asking those questions to get that information out of the one that called. First of all, we looked into the technology that we would need to build. At that point in time, the technology within machine learning and so forth was in its early days in terms of building good technology that could understand what is being said inside audio, and do it live, and to a good enough accuracy to actually be meaningful. So we did analysis on the research field in terms of all of the publications out there and so forth, and we found that we believed to have a good methodology that we could follow, first of all. Then next, we needed a data set, we needed to be able to prove it on a data set somewhere that this methodology would work. So we started looking into the space of these critical conversations that we were onboarding first, and we did analysis on all of the clinical research within that field, found out who did the best clinical research, and then we started hammering on the phone here, starting to call and ask them whether we could try a new methodology on their data. And fortunately, the

first call that we made, they were actually welcoming an innovative technology. So we asked for some data for us to prove our technology, they gave us the data, and then we proved that our technology could actually be capable, even in its infant form, to make a change inside these dialogues. Our go-to-market strategy, first of all, was that we need to prove our capabilities on a data source that was amongst the best in the world, and we quantified the best-in-the-world part from the clinical research. So we said, if our customers did a lot of clinical research within the field and got a lot of citations, etc., then we believe that a healthcare provider should be one of the best in the world. Our go-to-market strategy from that point was basically to go to all of the other best-in-the-world healthcare providers within that field to actually gather data. Because gathering that data from the best-in-the-world, we could make our AI learn from that data, and thereby make a really vital technology for the ones that are not doing as well. So from that perspective, we started aggregating all of the data from the best-in-the-world, learning the models, the machine learning models, to become better, and so on. And then after that, we went to market towards the second-in-class, third-in-class, and so forth, and started aggregating and harvesting their data, showing our capabilities on that data, training, learning from some of it for our machine learning models, and so on. So we started out in Denmark because of the Scandinavian healthcare system, first and foremost. And the language barrier was definitely a problem, because only 5.8 million people are speaking Danish, right? So building a technology that is definitely language-dependent is not always the best outset to build that for a population of 5.8 million people. However, the advantage of building it in some of the best healthcare systems was superior to that. So what we did is that we onboarded our first customer in the region of Copenhagen, and then we took the plane all the way to the West Coast in the U.S. and onboarded our second customer, and that was a U.S. customer, to make sure that we could actually tackle the problem of starting a product from a language minority. In order for a business like ours to scale properly, we need to build as generically a product as possible. And from the outset, we built our product to fit into the workflow of the general healthcare institution. And that means that we are actually live with the product, spanning from Sweden to Denmark, all the way to the U.S., having a lot of U.S. customers, that's our biggest market, with the same solution across all of these countries. We are covering approximately 50 million patient interactions a year right now, and we will be at approximately 100 million by the end of this year. We started out in the very critical conversations, and what we have proven in the last year is that our solution can be used across all of the different conversations in healthcare. So right now, our solution is spanning across the very critical 9-1-1 encounter, to the health plan provider, so that's when you call your health insurance provider, and to the nurse lines, that's similar in the region of Copenhagen, it's called 1813, but there are nurse lines scattered all the way across the U.S., to the hospital, communication centers, all the way to the physical dialogue, like the one that we're having right now. So our product, what we've proven in the last year, is that our product is actually applicable throughout all of those different workflows, with the similar product across the workflows. One of my biggest learnings in this journey of ours, until now, is that being true to the fact that I'm personally from DTU, and being true to the fact that I'm really good at understanding subject matter, and I'm really good at analyzing. My PhD has really worked in my favor, in terms of actually being quick at scanning our information,

and being good enough at questioning what I learn, and questioning to the right people whether I understand it correctly. When you know more, then you feel you know less, right?

--- End of DINK2.5 corti_case_final (Original).txt ---

--- Start of DINK2.6 paul_pop_-_ressource_management_final (Original).txt ---

Hi, and welcome to this segment on scalability and virtual resources. Our world is becoming increasingly digital, physical objects become virtual, they become software or data sitting somewhere in the cloud. And a key differentiator of businesses is the ability of a company to handle virtual resources and scale them based on demand from users. In this context, engineers like yourself need to understand the difference between virtual and physical resources, how to achieve scalability and manage these virtual resources, understand how to handle some of the disadvantages that come with solutions like cloud computing, and how things like networking effects create value through adding more users. Software is eating the world. Objects become data and software sitting somewhere in the cloud, music that we used to listen on a Walkman or a compact disc is Spotify, movies are Netflix, everything is becoming virtualized, it's data and software. Even in a factory, things like programmable logic controllers that control robots and gateways that provide communication are virtualized, they become software sitting in a data center and then you control the robots from the data center. This is happening because it has many advantages, scalability, cost savings, flexibility, it could even help sustainability. This is a car, it's a physical object, right? And you know what they say about cars, a car loses half of its value when you go out of the dealership. And that's true, it has maximum value when you buy it and then as you use it, it depreciates, it wears out, it has rust, it becomes less valuable. Now contrast that with a social media platform like Instagram, where when it has been released it didn't have that much value, but as more users are using it and upload content and create connections, it becomes more valuable, value increases when more users are using it. Unlike the car, Instagram becomes more valuable with use. These are called network effects and the value of a product or service increases with use and network effects are important when you design digital products and allows you to scale rapidly and enter markets quickly. This is not a car, it's a mobility service. Instead of selling cars that depreciate in value, they provide access to vehicles via an app, the green mobility app. We noticed with digital innovation this shift from product-based services to a service-based model, from products to services, and this is a fundamental component of digital innovation. This also allows you to have additional revenue streams, green mobility has probably a lot of data about how their cars are used, about how users, drivers are using their cars. They could for example sell data about battery management to companies that optimize that and this shift is enabling a lot of innovation. Behind me there is a data center at DTU that helps scientists to run more experiments by scaling resources available. Scalability is the capacity of a system to handle increased workload by proportionally increasing systems resources. A classic example is the Amazon website Black Friday before Christmas where you have to handle a lot of e-commerce. As a business, if you are worried about handling a peak workload, you have to buy more servers. With cloud computing and a

data center like the one behind me, you can temporarily increase resources to meet demand. This is possible because of virtualization. I can scale because software and data are immaterial, it doesn't cost me anything to make a copy of data or start software or a new machine. There are many advantages with cloud computing. We talked about scalability, this is also called elasticity, to quickly match resource capacity to demand, to make it elastic, increasing resources during peak times, reducing during off-peak times. A big advantage is cost. A startup or a business can access a lot of virtual resources without investing in expensive infrastructure. You pay as you go, you pay per usage. One challenge of cloud is that it uses different programming models. You have to use maybe a microservices architecture, a collection of loosely coupled services allowing efficient scaling and isolation of failure. You have to use containers, you have to use orchestration of containers. Containers encapsulate the software in a complete file system that contains everything needed to run on a new virtual machine. And orchestration helps you to manage and scale these virtual machines and containers. But with the cloud you also have disadvantages. Sometimes the performance can be a problem. Virtual machines all share the same physical hosts and maybe that's oversubscribed. You may have a problem with network latency. Some users may be far from the data center and it takes a lot of time for data to travel long distances on the internet. It could be a problem for high-performance real-time applications like, for example, some video transmission of a live event. Here you can use models such as queuing theory to evaluate the delays in your network. You can use load balancing to evenly distribute the workload. You can use orchestration to improve responsiveness of your applications. I mentioned the cost as an advantage, but cost can also be a disadvantage. Costs can escalate quickly if you are not careful to monitor and manage the virtual resources that you use. You have to optimize cost. You have to make sure that you only use what's needed in the cloud, shutting down unused parts, for example. Cloud also has disadvantages of security and privacy because you have multiple tenants, multiple companies using the same physical architecture. Although cloud computing companies make a lot of effort to separate these things, sometimes there can be data breaches. Another disadvantage is resilience. What if something fails in the data center? You can address that by, for example, building redundancy into your service by using multiple data centers in multiple geographic areas. This can improve fault tolerance and it can reduce downtime, increasing availability of your solutions. Computing can help with sustainability. We don't have to kill a tree to read a book. I can read it on a screen. But data centers and digital devices also consume a lot of energy. Around 2.5% of global emissions are coming from data centers and digital systems. This is actually similar to air travel, and we're all worried about air travel. What we need is green computing. We need to reduce the energy consumption of data centers, of digital devices. We should use renewable energy sources like wind or solar to power our data centers. And some ideas are, for example, to reuse the heat generated by data centers to heat homes during winter. We covered several topics, from scalability to virtual resources, how to use cloud computing in the right way, how to create value through users, and so on. I hope I convinced you that scalability and virtual resources are an important topic. It's essential for you as future engineers to use it and apply it in a way to drive a successful digital innovation.

--- Start of DINK2.7 christian_d._jensen_final (Original).txt ---

Hello, my name is Christian Lamsgaard Jensen and I'm an associate professor here at DTU Compute. Today we are going to be talking about scalability and some of the security issues that arise when we build secure systems or large scale systems. The ambition here is not to make you more proficient as programmers because you cannot learn that in just a few minutes but hopefully you will learn some of the issues that you need to think about when you build large scalable systems from the security perspective. So in general there are three main areas that we need to address when we build secure systems. The first one is that as systems grow they become more complex and so complexity is an issue that we need to address and a problem that generally arise for security in these large scale systems. The second thing is that the attack surface, the overall area from where the system can be attacked grows when we build larger systems. And then finally as we build larger systems we build them because we want to support more users so the entire user base grows and therefore the user management becomes a very important issue that we need to address in large scale systems. So those are the three main areas that were important for building security in scalable systems. Complexity is an inherent property of the systems. As systems grow larger they typically also become more complex because there are either more components in the system or there are more instances of the same component inside the system. So that also means that we need to manage all these different components. We need to orchestrate them so that they work in concert together to support the purpose of our application and all this extra interaction, all these larger number of interactions and APIs that we need to support inside our system means that the entire system becomes more complex so it becomes harder to test and to verify in different ways. So that is one of the problems that we need to address when we build secure scalable systems. This means that we need to think about very much about and understand all the components, all the security properties of our components, all the interactions and the security properties of these interactions and potential any side effects that could be there either inside the components in the interaction between the components or between the system and the outside world. So secondly the attack surface. The attack surface is an external property of the system that describes how we see the system from the outside. As systems grow larger they will have presence in more locations, they will be accessible in more ways from more people and therefore there are more ways that things can go wrong essentially inside the system. So this number, the increasing number of locations, the ways that we have replicated our system throughout the world but also the APIs in the systems that become accessible from more peoples and from more locations is something that we need to address. Just like the Germans faced during the Second World War with the Enigma machine where there was a machine that was captured by the British from a U-boat in the Atlantic and that way the British knew more about the internal settings of the machine and helped them decrypt the Nazi communication during the Second World War. So what you need to take away from this is that we need to be thinking about what kind of interfaces we expose to the outside world, what kind of services we make available and to try to keep this as small and as compact as possible. And so the third property we need to think about is the user base. That when we have large scalable systems then we get a much larger user base and all these users have to be enrolled into the system so we need to find out who they are and give them some kind of a

system identity. We need to be able to authenticate them for instance with a password when they come back into the system so that we can treat them in the same way whenever they are here. And all these things are difficult in large scale systems. So imagine that Facebook actually had to verify your real world identity before they could give you a Facebook account. This is something that is clearly not feasible when you have billions of users in the system. So the fact that you have all these many users creates a problem. It becomes easy for people to create fake users who can distribute fake news. But also you have the problem of managing all the privileges of these different users inside your systems. So there are some users who have some privileges, other users have other privileges and you need to make sure that you have assigned the right privileges to the specific user that you have in hand because otherwise the overall security policy is likely to become validated. And once you have assigned these privileges, the status of users can change either because they get promoted in some way in your system or because they simply cease, stop being users in your system. And so you need to be able to change these, manage these privileges throughout the lifetime of the system. And that is something that has traditionally been very, very difficult and is still something that is very, very difficult. So what can we take away from this? I think the most important thing is that we manage our users in such a way that the consequences of their actions is proportional to the verification process that we had when we enrolled them into the system. So if we enroll people with very, very little confidence in who they really are, then the consequences of their actions should be relatively small. And that is something when we build systems that can be very, very hard to anticipate what are the impact of what users do in the systems. Basically, we need to have thought of those before we created the systems. And that's, of course, difficult. So we have now looked at the three factors that are most important to consider when we build security into our scalable system. The complexity of the system that we need to address to make sure that we manage it and that we understand all the complexities inside the system. The attack surface of the system that we need to also address and make sure that we understand where our system can be reached from and also where it can be attacked from. And then finally, the large user base that we get when we get a large scale distributed system and how to manage that large user base. Those are the three important things that we need to consider when we build scalable systems. So you have not been given any tools or prescriptions on how to build these systems, but at least you have been given some areas to consider and some places to look when you try to design your systems and implement them and in the future test your system. So enjoy building large scale secure systems.

--- End of DINK2.7 christian_d._jensen_final (Original).txt ---

--- Start of DINK2.8 podcast final.txt ---

In this podcast, we will unfold some of the perspectives and dilemmas associated with scaling of digital solutions. The podcast is designed as a complementary component to the videos in the module, and its intended use is slightly different. Our intention is not to have you scribbling notes while listening, but instead, we invite you to immerse yourself in the many discussions of this podcast while working out, or perhaps cleaning house, allowing you to contemplate and reflect on

various concepts at your own pace. Throughout this podcast, you'll be introduced to the following discussions. Security and user experience are the two opposites. Delving into the dynamic relationship between security and user experience, are they truly contradictory, or can they coexist harmoniously? Data and security explore the intersection of data utilization and security considerations. How can we harness the power of data without compromising vital security measures? Managing complexity and users uncover strategies for managing complexity while keeping user needs at the forefront, balancing intricate systems with user-friendly experiences. Designing a system that users can trust. We uncover the principles of designing trustworthy systems. What factors contribute to user trust, and how can these principles be applied? Can you scale in a sustainable way? Can you scale your innovations in a sustainable manner? We'll explore the intricacies of growth and sustainability in the digital landscape. Preparing for rapid scaling or not. Is rapid scaling always on the horizon? Delve into the considerations and strategies that come into play when preparing for or reacting to sudden growth. Hardware's role in scaling. Is hardware a limiting factor in the quest for scalability? We'll dissect the relationship between software innovation and hardware constraints. Learning from data while minding ethics. As we harness the power of data, we must also navigate the ethical landscape. Explore the nuances of responsible data utilization and the importance of ethics. You can choose to snack-size this podcast and listen to one topic at a time, or just listen to it in its full length. Enjoy. So, welcome to this podcast. Today we are going to unfold some of the dilemmas and discussions around innovation and security and safety and scalability and issues that sometimes pop up. But we're not always conscious about that in the early phases of innovation. And I'm really happy today to welcome Tim McAloon, Liene Clemmensen, Christian Damsgaard and Paul Popp. And I will let you introduce yourself briefly to your background. So Tim first. Yes, hello. My name is Tim McAloon and I'm a professor of sustainable product service systems at DTU Construct. And what that actually means, my main focus is sustainability, but a means to that is the so-called product service systems or alternative ways of providing value than just products and product development. And that, of course, includes things like digital solutions. I think that's why I'm here today. It is. Thank you. Welcome. Liene. Yes. Hi. I'm an associate professor at DTU Compute in machine learning and data-driven innovation. And so most of my research is in the machine learning methodology, but also how we can actually use data and the right data when we are building those digital models. And what are we building for? Is it a population-wise or individual target? Welcome to you as well. Thank you. Christian. Yeah. Hello. My name is Christian Damsgaard-Jensen. I'm an associate professor in the cybersecurity engineering section where I work on cybersecurity, interestingly enough. So, yeah. Welcome to you. And Paul. And yeah, I'm Paul Popp. I'm a professor of cyber-physical systems, but I'm actually a computer scientist. I design algorithms and software that help you analyze and optimize real-time applications that are distributed across the computing continuum from cloud to edge to IoT devices. And here, networking also plays an important role. So welcome to you also, Paul. And I can maybe say that my name is Pierre Beckor. I'm also an associate professor here at DTU Compute. And I'm working on user experience as one of the parts here, and also how this applies to innovation. Sometimes, at least when I've been in large companies, we have divided the functions. So there's somebody who's taking care of the product management side and the user's experience. Others are taking care of the backend and the designs, and somebody is

then taking care of the security and so on. So we end up with a little bit of polarized world, perhaps. And sometimes we also see security as sort of the enemy of the user experience in the sense that things become more complex and more sort of difficult to work with. Is it really like that? Or what is your perspective on security when you design things? I think it is, in some respects, correct that in order to authenticate people and find out who we are actually interacting with, then things become a little bit more cumbersome. But also, it's often because we haven't designed systems for people or for ordinary people. We very often design our systems for engineers to use, because we are engineers ourselves. And therefore, it can be difficult for people to understand the security abstractions and the abstractions of the applications and how to interact with it in a secure way. And sometimes it's not even really possible. So one example that I often give is the message that we often give people is not to click on links that you receive in an email. And then most organizations have some kind of workflow system to manage their finances, which means that very often they distribute emails. And then you can click and authorize a payment in that workflow system through your email. So essentially, we're telling people, do not click on links in emails, except the links in emails that you have to click on, which is not a simple message for ordinary people to understand. Then we can simplify that by saying, OK, but typically, malicious people are from the outside. That's at least the way we like to see things. So don't click on links in emails coming from the outside. And then three months later, we have an employee satisfaction survey where we have an external company providing us with a survey that we have to click on and access. And so it becomes very confusing for people to relate to this. So the security hasn't been thought in a methodical way from the beginning of the system. And therefore, we are fighting our abstractions as we're going along with the system going forward. So I mean, security obviously has a benefit for the company that wants to protect some data. But isn't there an issue also, for me as an individual, that I don't do anything stupid, at least I'm not supposed to do that. So what's the added value for the individual in the organization? So I think both in the organization and the organization's users or customers, I think security has a very important role to play. And for the customers, it is, do you want to engage with an organization that has a track record of not protecting your data or is abusing it in all sorts of nasty ways? And of course, you don't really want to interact with these companies, which means that if you don't have a sufficiently good security, then you might not have customers at all. So that's kind of like the customer perspective. For the employees, I think if you're working on a product, it's not so much your own. I mean, of course, your employee records and things in HR is something that you'd like to have protected. And also, of course, the crown jewels of the company, all the intellectual property or whatever it is that you're sitting on, it's important to protect because otherwise the company will not be there tomorrow. But apart from that, I think the individual employee is probably only concerned with security insofar as they live with it. So they have to authenticate themselves, multi-factor authentication, have a device and all the things that the hoops that we are asked to jump through. And then, of course, they have to implement it in their products. So they need to understand it, at least to some degree, that they're not doing something stupid. And then, of course, for the companies, they want customers, so they need to be seen to be secure. They probably don't want to have some of these fines that are going to be introduced with the EU regulations that are both in place and on track. So also, therefore, they want to

improve their security. And then finally, of course, I think there are business leaders who just want to be good people and therefore they want to be secure and secure the data of the people that they are recording in some way. So, Christian, you talked about value also here. Tim, from a more business perspective, do customers typically understand the value of security, especially when they need to deal with a little bit more cumbersome?
product and a double login, something multi-factor authentication thingy? I guess so. So I think there's different types of ways of creating or transferring value from a company to a user or a customer. And we typically work with three main modes, there's the sort of results and performance-based value creation, which is where the customer is less involved with the interaction of a product or with the data themselves. I guess the security issues there are less front of mind for the user or the customer. Then we have access and availability-based value creation or business models. That could be, for example, your ShareNow or your Scusa in the tower. Depending on how often you use it and depending on the type and level of interaction, I think there's some security issues there that you may think about. But I think when the interface comes onto your mobile phone, as soon as you got past that phase, I guess once you've authenticated yourself, once somehow been approved to be part of the club of a ShareNow system or whatever, then I'm guessing that the security mindset is less front of mind. The third type of alternative way of creating and transferring value from a company that we often categorize as these so-called product longevity type services, so services where we try and keep a product on the market for as long as possible through remote maintenance. These are often business-to-business types of services. Then I know that security is a big concern because all of a sudden we're operating a crane for a company in a mine and we'd like that crane to function, but we'd also like it not to malfunction. And I know that the security issues there and the security concerns have raised over recent years. Windows more maintenance of its product life warranted services, which are being the data which is transferred back and forth then. So I'd probably say there's different graduations of concern. Data and security. So we kind of become accustomed to sharing a lot of our behavioral things with companies. That's what recommender systems are based on and you touched on that, even the car sharing system can actually reveal very highly sensitive information about us as individuals. So of course, we're just not thinking about that very often. We just share it freely and that's value, of course, for the company. It could also be value, but it could also be a tremendous disaster for the company or for the individual. But that's part of the value creation and that's the data thing, Lene. I mean, this is the value that might be inside of data, which could be the personalized ads that you're saying. It could be, how am I using my phone? Do I remember to charge it every night, etc. Then I'm a responsible person, so I might grant you a loan based on your phone usage or just the behavior you have. It could be the variables and again, the behavior we have there and we can sort of then notch people to a healthier lifestyle. So there are many possibilities, right? How can we use those data in order to bring value? And that could be depending on the company you are, what do you think of as value, right? Yeah. So what we do and how we interact actually creates value, but it can be, as you say, highly sensitive value. I mean, some of what you said about sleeping patterns or other things can be maybe something you don't want to share with everyone. So having this security mindset or understanding that there are perspectives of that is important. If I look at some of the things that are maybe sometimes happening outside in the big

world, it's also about data leakage or things that are sort of disappearing. You mentioned the example, Tim, of older systems that go into maintenance mode. And I mean, even though you talk about the service by design, maybe that things have to be designed from the start, what about the end? I mean, there are a number of ways of talking about the end of a computer system or an application because, I mean, we're seeing things just being discontinued and then you're stuck. Hopefully data that belongs to that application will be deleted after that point, except that there are probably legislation saying something about whether data should be retained for a certain period of time. I believe that you would probably also argue that you would need to keep it for a certain period of time so that people, according to the GDPR, could claim to get their data out and use it in another application that they might want to switch to later on. And then once you have that kind of application perspective and the data from the applications perspective, then there is what you actually do with the physical hardware. And I think that most companies these days are conscious about wiping hard disks, hard drives properly when they decommission their computers or reinstall them for other purposes. But in the past, there have been some nasty cases where they basically just shipped off their computers to schools in a less developed country or something like that. And then application data was available for school students in other countries. And there might also be possibilities of innovation saying, okay, you can keep your data on your phone. I'm just going to send an algorithm that computes the value right there distributed. So there are lots of innovation possibilities in thinking security and data protection into the systems in a different way. Yeah. So it could also be done in more clever ways, perhaps. So we don't necessarily need to store all that kind of data on cloud computers or whatever. So that's one of the things that are, at least from a security perspective, quite interesting in some of the development in artificial intelligence at the moment, this whole idea about federated learning, where we actually keep our data decentralized, but then we distribute the algorithms as that. So function shipping rather than data shipping. So federated computing. Federated learning. So it seems like, based on what you're saying, that as systems are being used, we also build up more and more value, sort of non-tangible assets maybe, but the data that we have, that even if the business itself might end, or the original business idea might terminate, then there could still be value also in bringing that on to something else. And that kind of brings us to this perspective of where the data is stored. And you said, Christian, that most responsible companies, they decommission data in a good way. But then we move to the cloud. Yeah. The data can sit in a cloud and not on the hard disk that you wipe. And we often say that data and software are immaterial. They are bits, ones and zeros, sitting somewhere on a hard disk, traveling on a wire, being executed in a CPU somewhere. And because it's immaterial, you can see it as virtual. You can see that the data and software are virtualized representations. And as a developer, if you want to develop applications, nowadays it's relatively easy to get access to very large virtual machines in a cloud or data centers and just start your services there. And you, as a startup, as a company, you can just pay for the usage of those resources. And this allows you to scale applications a lot. And there are challenges related to this. How do you rethink your programming models and the data representation to reside in the cloud? You have to program things differently, probably as microservices, if you want to run them in the cloud. You have to think about where your data is located and how quickly you can access it. Because

depending on how much data you need to access, it may be residing on different computers, different hard disks. And actually, it takes a while to gather it together and work on it. And as engineers, we are often falling in love with some technologies. And when people develop new solutions, they often want to emulate companies, I don't know, like Google and Netflix. Let's all use Kubernetes to scale things up. But that's just too much complexity. When you start up with some ideas, it's better to just have a simple product to judge product market fit and have a very simple solution that is not necessarily extremely scalable. But be worried about getting a lot of users later. Initially, be worried less about the scalability and more about your product market fit, about the business value of what you do. But of course, you may be lucky and you may be getting a lot of users. And then you run into different problems. We just said that you can have a pay-as-you-go model with the cloud, but it does not come cheap. Sometimes those costs can balloon. And you also have to be aware of how you use cloud resources. And a lot of times, data does not reside only on your local hard disk and in the cloud. It can be anywhere in the network. There may be applications where you need real-time response, thinking about video or maybe some real-time video, for example. Or maybe you want to process data locally and not in the cloud for all sorts of reasons, maybe speed or privacy reasons. Again, we're talking about cloud. coming closer to the edge of the network and then we call it not cloud computing but edge computing and then data can be anywhere in this it's called computing continuum from your phone or IOT device to an edge computing server for example or data center somewhere traveling in the network in a cloud data center so and depending on the type of application that you develop you have to be aware that yeah what's the best solution for for what you want to deliver managing complexity and users so I mean in the clouds you might have sort of centralized user management I guess but but when it comes to sort of these very distributed systems where beta can be almost anywhere is it then more difficult question do you think in terms of managing the the complexity of users I mean often when we do a minimum viable product we might not even want people to sign in initially until they start using it for a while so what what is your perspective on that I mean is it is it possible to design it in already from the start or should you just accept that well you know we care about that later that's a really good question because I mean managing identities in a very large scale if you cannot bring them on to one common platform which would be what what you would do if you have managed your your own user database somewhere but you want to run it this more decentralized model and also if you want to have like an increasing number of government say well we actually need to have age verification of people logging into our system so or to your systems then you need to link those identities that people create in your application with real-world identities so they can verify their age at least something else on the outside world that that can attest to their age which means now that you need to to relate to to the different identity schemes in different countries which makes kind of like global scalability a bit of a challenge because they are very very different around the world and the legal constraints on what you have there is very very different around the world so the complexity of the the overall kind of identity management problem is exploding if you want to go global. How to design a system that users can trust? So that kind of I mean brings up another topic perhaps that that I thought about in this context which is trust and of course as individual users we need to trust the systems but but if you start relying on other companies that are providing maybe even service identities Google

or the large players that you can always sign in with Google or Facebook or whatever how do you mean from a from a design perspective when you when you're trying to figure out how to set up your platforms I mean do you need to trust your cloud computing provider I can just go with the cheapest one and then because it's it's all in the container so we don't really care or how do you see that? It's a big deal I don't have a lot of experience but when you develop you have to rely on libraries you have to rely on other providers there are many attacks that can happen because maybe somebody has tampered with some open source libraries or some repositories where you you get your packages to manage your docker containers and so on so that's definitely something you should be aware of and you should have used best practices that minimize the risk of tampering with your development pipeline. So if I was a startup and said okay I need to get going here I'll find a solution I think I've designed my system that I need some cloud services and so on and I come to you and ask for best practices from a perspective of designing the system I mean making making your decisions and where to put things I mean... Yeah definitely security is an important perspective but I'm not I'm not considering that you know as researchers you often look at your own little topic and whenever I am thinking about designing something that scales I don't consider security but what I do consider is the security impact on safety because we often think of these two things separately but security may impair safety I work a lot with safety critical systems for example think about medical devices think about an insulin pump or any kind of pump delivering some intravenous for example liquids in a hospital they have to be safe and you have to certify them for for safety of course but any security breach can actually impair the safety and there is a tension between safety and security with security you want to patch as often as possible a new patch would maybe close some security gaps but with safety whenever you change the functionality through a patch or an upgrade you're actually introducing safety risks and you often cannot dispatch because you need to recertify so these are it's a tension between safety and security. So in terms of this trust discussion that we had also now you brought in also the safety from a data collection perspective also and from sort of the machine learning perspective of getting started with something how do you see this I mean I guess we agree that it's important to trust the services that you build or to make sure that your customers will trust those right? So I think from a machine learning perspective is also really important that they are fair and are not discriminating and these are kind of the things that we would need to verify and check and they're very much it's also related again back to data what kind of data do we collect if we haven't seen any kind of training samples that reflect specific groups of our population then our system will not work for those groups of the population and they may even become very discriminative. So for you I mean trust is also about making sure that you design for fairness and having representative data and what you actually collect. And ethical use right so what do we want to use it for is this does it bring value and does it bring value in in a sense maybe it's a very targeted population but if that somehow can bring a dissatisfaction or some kind of stir for other parts of the population this is something we might think of also right to just consider what what are we actually doing sometimes yeah this can be difficult because you're very focused in the beginning you need to produce your MVP and be very targeted on your users and you should be to make a good product and bring that value to market and then what about the rest right. So if I would ask you also about good advice for my startup idea my innovation idea in terms of the data then I mean what what would be some of the

keywords I mean you already mentioned this trust and fairness and in terms of the relationship you have with the users and what you collect. And in collecting the data it would be covering use cases covering your population right so instead of just thinking of collect as much data as possible convenience sample you would need to think very carefully about cover covering that population and the way you sample it's really important. So maybe design your initial systems for collecting the data that you actually want to create the value on and not just any kind of random stuff. Yes. Can you scale in a sustainable way? Tim from a kind of also I mean one of your perspectives is also the sustainability of things and of course that comes with different aspects here but could you just maybe unfold a little bit about that I mean I hear the word cloud computing I'm thinking about you know lots of computers that are standing somewhere in a basement and are generating an enormous amount of heat and just dissipating that into the air and it doesn't sound super sustainable from my perspective. Yeah and there's many studies about this and I don't have the the numbers and the data to. I do it's 2.5% of the global emissions are generated by ICT. It's on par with air travel. Exactly. We're all you know the flick scam or how is it or all flight shamed and not to take the airplane but everybody's using chat GPT which is a ten times more energy consuming than a regular Google search and then and then how can we reduce the energy consumption of and I'm reading somewhere that the Chinese who are now deploying a lot of 5g they turn it off at night because it consumes enormous amounts of power I'm not an expert on that but it's also the communication not only the computation. Right so I think it's the we've created an iceberg of overhead that is really difficult to pinpoint on one particular intervention or one value exchange because we have so much data out there and so much data ready for us to be able to use and what I didn't have the data on but I was gonna that it actually exists is some estimates of about what is the eco footprint as we call it the environmental footprint of a one tune on the CD versus one tune on Spotify or any other service like Spotify for example and intuitively we think it's better to have to not own the physical product right but how many copies of one song a particular song are there out there stored on how many servers and how many backup servers and how many backup backup up servers, and how often are they being updated and so forth, and where in the world are we putting them? The colder parts of the world? Because it's nice and cool, it costs less to cool them. Great idea. And so we start to heat those parts of the world up. So there are some problems and challenges there. I think one fix to that, of course, is technology to compute faster and to compute better. But we often create these rebound effects that because we compute faster, we don't... there may come a period where we compute more effectively, but because we compute faster, we actually do more with that faster computing power. So we actually end up not just solving the problem we were hoping to solve, but actually we create more stuff to solve. And I'm not a techno- negative person, but it's just a fact that we are creating bigger and bigger environmental footprints with what we're doing with data, and not only storage and computation of data. Yeah, I often think as well when I see these self-driving cars, how much do they actually take of energy to actually to compute the cameras and the sensors and all the other things to be able to do what a human being can do on a couple of pieces of rye bread? It's quite a lot. So we should have not only Federer and Le Bruni, but also rye bread computing, maybe. That's the comparison unit, right? Because I mean, it's ourselves as humans versus something else. And I'm all for automation. I'm all for many different smarter ways of doing things to a point. And I think we need to ask ourselves where

is that point with every application? So basically you're saying also that it's similar to when we built more lanes on our roads, that we just see an increased amount of traffic there. So perhaps it's not just about adding more computers or faster computers. But I mean, your perspective I think also was that perhaps we could also think about how we distribute our services. It's the whole topic of green computing. ICT, information and communication technologies, they can both help sustainability or they can help us, I don't know, get more oil and gas and then impair sustainability. So how do you use it in a way that helps sustainability? And how can you reduce its own resource consumption? You can use digital technologies to optimize the energy grid, for example, and district heating. And you can design new infrastructure and new ways of distributing data and computation to reduce consumption. Here it's very important to use renewables to power data centers, for example, and use their excess heat to heat homes in winter, for example. So I think as engineers there's a lot of room for innovation that helps us use digital technologies in a way that helps sustainability. But here it's very important the role of policymakers. You need policies that pushes innovation towards areas where digital technologies are helping and they are not actually impairing sustainability. Preparing for rapid scaling or not? So I think, Christian, you talked about in Italy that security is something that you need to do early on in the way, but design perhaps. What about scalability? Do you need to think about that when you dream about your MVP also in Italy? I think you should be aware there is an issue and if you get a lot of users then you should have an idea of how you can scale, but you shouldn't over-optimize. You shouldn't from the beginning think about scalability. You shouldn't emulate, like I was saying, what Netflix is doing and what Google is doing. You shouldn't get from the start to use Kubernetes and so on. You should, of course, have a solution that can scale, but be worried about large-scale scalability, let's say, later on as you get more users. And you can also reach the other end of the spectrum. There are companies that have so many users and they're consuming so many cloud resources that they have bills of, I don't know, millions of euros per year. We were discussing before we started about this example of Basecamp from 37signals. It's a productivity suite and they switched to their own data center. Basically, they realized it costs less to just pay Dell to ship a pallet of servers to a data center where they install it and hook it up to the Internet and then they can start their services there in some simple way. And they could reduce their costs by actually managing their own data center. And there's a lot of software and tools that allow you to do that, not at the scale that Google is doing, but at the scale that can serve your users. So that's also something to think about. How can you right-size? How can you pay only for what you use? Cloud computing can be expensive. So just to add on to that, there seems to be a trend that we are accepting new technology more rapidly and kind of like internalizing it more quickly. So if you're looking at some of the bigger companies and how long it took them to get 100 million users, I think I saw some numbers recently where it would take Google Translate about 52 months or several years to get 100 million users. It took ChatGPT two months. And we're seeing that TikTok also a few months to get to that. So while I generally agree with what Paul just said, you need to realize that if you have something that is a brilliant idea, then perhaps you only have like a month or two to upgrade to a more sustainable, in the non-sustainable way, software platform to service all the many customers or users that you might actually get. Yeah, of course, you should dream big. You should dream about becoming ChatGPT that gets 100 million users in a month

or so. But that's very unlikely to happen. So you should not over-engineer your solution. That's what I'm saying. You should definitely be aware that scalability is important and have a solution that you can scale up to some degree. But you should not spend all your time in worrying about being swamped with a billion of users, because that's unlikely to happen. Of course, it can happen. And of course, you should consider in your design that you need to scale. And by using cloud computing, you can scale. But at the same time, that should not be your primary concern. Your primary concern should be serving the users, making sure you address their needs, creating value for them, making sure you protect their data and privacy. And then using something that is, let's say, state-of-the-art and that allows you to scale, but not go beyond that, not think about being swamped by a billion users the day you open your service. Even ChatGPT had some issues in the beginning, right, of scalability. And Microsoft helped them to scale. They signed a contract, 10 billion US dollars. How much was it? That's a good start. If you become so successful, don't worry, Microsoft will knock on your door and help you scale. So basically, you're saying that we should, you know, prepare for what we think is the most likely scenario, but then be ready for when the best thing happens. Is hardware a ball and chain when it comes to scaling? I maybe ask a question because I come from the product world, the physical product world. And there we have different, let's say, different advice to first students here, but also to industry about where to, what different strategies to have at the different stages of scaling a product-based business or a product in itself. In the software world and the data world, do you have any particular strategies for what types of scalability, affordability you put in at a cold start versus when it gets a good rematch to the inflection point where it goes crazy, but also from a security perspective, are there things that you say you should design security in from the start? And then you're saying, well, that you should not design, over-design things, but are there some good practices in different stages of your business? It's basically another way of maybe thinking about this. Is there continuum or do you need to sort of plan for some transitions between the different stages there? From a data perspective, it's easy, right? You just have a bigger server. Yeah, I guess. But I think also the big companies, they're always looking for data source n plus one, right? Yeah. And they're always looking for data source n plus one, right? And also the big companies, they're always looking for data source n plus one, right? What's the next thing I can collect on you that will give me value down the line? They're really seeing this as a business model, if you think of Google, et cetera, right? It's really our data that's their value. And then I think they have thought about this for sure, right? And also how can you then... So one thing is just storing the data. Another thing is making it accessible for the digital products so you can actually compute something on it and bring that value back to the consumer. Yeah.

From a security perspective, Christian, if you look at these different stages, what would be the most important things? Maybe Paul also teached you about what are the three important things that we should consider when we design things. When you're on the scalability journey, then as systems scale, there are going to be more access points, so there are going to be more people engaging with your system from all locations around the world, so you need to have a local presence in more locations, which means that the attack surface of your system becomes much larger. And that's something that you need to consider. And then also, once your system grows, then it typically also becomes more complex. And so when you build your

system, try to engineer it in such a way that with scalability you don't get added complexity, which means that relying on technologies where essentially you can get somebody else to manage the scaling for you, so that you don't need to introduce new levels or new hierarchies of services and caches around the world, but to see if you can't just have the same uniform service from your application point of view, regardless of where you are, and then pay somebody to manage the data centers or manage the data centers yourself, as Paul also suggested, but in a way that's uniform around the world and you don't get added application complexity, because otherwise it's going to be very, very difficult to manage. So from your perspective, I mean, it could also be done as a regional, geographical start-up time. You start with Europe, and once you fix that, then you can move to other parts of the world, for instance. And that's not unusual that people are trying to get to the local market first, maybe not so much in Denmark, because the local market is not really worth targeting alone. But generally, I mean, you will see a lot of things that are popular in one country, and then suddenly it kind of like spreads around to more countries. I don't think that's unusual. Although there are some cases, I guess, like we had MobilePay is one that started here in Scandinavia, at least, and there may be also other services that are originating here from Denmark that have been starting out as small ones that are growing, I guess. So that could also be a strategy. It happens. I also see it with, now I'm dating myself here, Skype, which used to be the WhatsApp of the noughties. And that was also kind of like a small app in Estonia and then Scandinavia that kind of like spread around the world. And they also got bought by Microsoft at some point. But you're touching on something. You talk about products. Those are physical. We talk about virtual world. I think a lot of innovation can come from having physical products, from having hardware. But hardware startups are extremely difficult to work with. Then you cannot just, you know, scale it in a cloud. Scalability there is about logistics. You have to be Tim Cook and you have to have Apple money to be able to design a good product, a physical product. If you want to design a wearable, for example, probably the solution you can get from suppliers is not the same that somebody like Apple can get because they can design their own production processes. They can buy the whole factory and invest hundreds of millions of dollars just to figure out how to integrate something that you can never dream about integrating. There I am excited about hardware startups and you should not be afraid of working on that. But scalability there is something else. Scalability there is about supply chains and about making sure you can source it from the right places. And then it's an issue of supply chain risk and how do you manage that. It can introduce security risks also if a supplier maybe puts a backdoor in your product. It's a whole different host of topics. But because products and hardware are so difficult, you have these stages on how you scale. Whereas with software, people perceive it as, OK, let me from the beginning make it extremely scalable. And we don't often think about stages. That's my impression. Yeah. So there might be stages there still, like you pointed out. But you put your software on the server somewhere and then it's just a question of whether you can handle the download charges and process the credit card transactions, right? And then you're done. Exactly. But I can attest to, I mean, I'm a bit involved in a startup that is doing a hardware device and that needs medical approval. And doing that in this time where you have supply chain issues is not making things simpler. So you're longing for the maybe less complex software world. But just to put a little bit of a silver lining on the cloud that Paul just introduced, then when Apple created the

iPod, it was basically a bankrupt company. So when Steve Jobs came back, that was essentially, he took over a company that was more or less bankrupt. And then with a few innovative products, he actually turned the whole Apple company around 25 years ago. So if you have a good hardware idea, don't despair, but think that you can do stuff. But I think that's actually a good point also, Christian. I think because the iPod was, of course, a device that could be sold and create value for the users. And there's a lot of interesting sort of points about how that was designed. But I guess that we're not using our iPods that much any longer, if we still have them. We've moved to sort of more cloud-based solution. But I think what has happened there was that because Apple delivered the iPod, they could also learn from our behaviors, what we downloaded. So even if it was like distributed computing, we would have an access device that contained the data that allowed Apple also to build information about how we were using data, right? All of these big companies are doing that a lot, learning from the data. It may not be just creating value, but just collecting them and then seeing, OK, what are the behaviors? Where are we going? What might be new market trends? Or how can you meet some of the, yeah, sort of maybe even unspoken user demands, right? Yeah. How can that benefit the users at the end of the day, of course? So at least I think that's an important perspective also. How do we ethically and in secure ways actually make sure our services benefit the users? At the end of the day, that's why we buy systems, I guess. Absolutely. And I think there's always a fine line between taking and using data to make service and value offerings better, and at the same time, going too close to the user and the user's data. So I think it's for sure an ethical discussion, and I'm intrigued to know how much that is actually being taught here at DTU about the ethics of data and the ethics of computing. In an innovation perspective, I think we need to know, and our future engineers need to know, where the lines are, at least to ask the question at the right time in the development program, the innovation process, and how close can we actually go there? And there's frameworks, and there's laws, and there's regulations that naturally put those types of frameworks into place. There's always new cases that we should have some form of a mindset of being able to weigh up value versus how close it's too close. Yeah. And I mean, maybe even in this time of chat GPT and large language models and transformers and diffusion networks and so on, that's essentially mining all of the data out there, everything that we thought about or wrote about or whatever. And of course, that brings a lot of other perspectives into this, that now we are maybe migrating to a slightly new set of challenges that we haven't met before, and we still need to apply some of the same methodologies for that. So maybe to sum up some of these discussion points, I think it's been super interesting to hear your perspectives on this, or you should have one piece of advice to give to somebody that is dreaming about starting the new chat GPT, or maybe a little bit less. We can also do with 100 million users in three months, that's also okay. So what would be sort of the key advice? What is the key takeaway that we should think about now? So do you want to start, Paul, about your perspective on that? We've been talking about many different topics and perspectives, and somebody that would like to start an innovative product or service may be worried about all of this complexity. My advice would be, don't worry, try to ignore it as long as you can, because that's how you survive. If the first thing you think about when you want to design something is GDPR, then you'll probably just be bogged by all of this complexity that we have, especially in Europe. My advice is not necessarily that you should move to the US or to the UK to

escape, I don't know, regulatory frameworks, but at least just serve your users, figure out what adds value to them, and make sure that you do it in a way that has integrity and protects their values. And then, as things progress, you will naturally be facing all of these things that we're discussing, and then come back to the podcast and figure out what's relevant. Thanks. Karsten, your perspective? Yeah, I'm not quite sure that I agree with what was just being said because I think if you ignore security and especially if you ignore things like GDPR and collect data for secondary use, for instance, then the entire business model can be invalidated by regulation. And so you'll just be a failed company and you will not provide value to your users, you'll not provide value to your investors or to anybody essentially. So I think it's important to understand these things and then to think in security from the beginning. So perhaps you need to think of your product as is this something that I would use even if my worst enemy was actually the supplier of the product or the service. And if you can agree to that, then you're probably okay. Nina? Well, I also think that the number one is understand your user. I think it's very difficult to bring any kind of value if you don't understand the user. And then, yeah, yes, I wouldn't ignore anything. I think this would be kind of if I start up a company, it would be the values of the company also to do something good, to be ethical, to be fair, et cetera. So it kind of, you would think about that all the time. But just because, yeah, I mean, it's, and then of course there are some decisions that you might push, like first you need to check, like, okay, can I even bring that value to the user? Can I make this technical stuff work, et cetera? And then you might want to think about the details of all of the other stuff, right? So you kind of do maybe prototyping and then prototyping, and then, I mean, the usual innovation pipeline, right? But yeah. Tim? Yeah, again, I think from my perspective as a product developer, innovation process person from more from the physical world, I bring this mindset with me that there must be stages, there must be developments over time of how to plan new startups, new activities, new value propositions. And I've not thought about it in the data world until today, but it's not quite true. I have a little bit of experience from a couple of platforms we've created from some of our research which are based on data and user data. And we've tried in one of them to think, how do we, from a scalability perspective, provide the right advice to the user when there's a couple of hundreds of users? And then what do we do when there's a couple of thousands of users? And what do we hopefully do if we hopefully get hundreds of thousands of users? And this particular project is in a stage where we have a few thousand users. So we're going from relatively static recommendations to doing cluster-based recommendations. So hopefully then we can do some machine learning, how can we sort of do that? So I think thinking those stages from a data perspective, I would say there must be a way of framing that somehow. I've not thought about the security scalability before this conversation here. I've always thought, well, we should make it as secure as possible because we're taking companies and people's data and we're living up to all the GDR and so forth. But how to design that in from the start, I think is a really important concern. At the same time, it's about getting bogged down and sitting, as Paul says, in the corner and thinking, my God, this is too complex for everyone to get started. That's the product development perspective. From the sustainability perspective, I think we need and must do things smarter to be able to somehow see if we can decouple the knowledge we create from the environmental footprint we're also creating. And that is an enormous issue. I just needed to get that word. That's important. That's also important. So

basically, maybe to sum it up, be an optimist, Paul, but don't be ignorant on what you're doing. Care about your users and the value you create for them and continuously reflect on your journey and be smart. So thank you very much for coming today to participate in the podcast. I think it's been very interesting to hear your perspectives and I think it's good that we sometimes don't bring the same perspective because the world is complex and we're engineers. We want to engineer things, but we need to understand the complexity.

--- End of DINK2.8 podcast final.txt ---