Lukas (00:00):

Time to start. So I'll get us going here and then in a few minutes I'll turn it over to Guy to show us a little demo of Typefi. But I'll start with just some quick introductions. So today we're going to be talking about how you can publish your accessible content automatically with Typefi. And we've got with us today myself, marketing manager, Lukas Kaefer, and we've also got Guy van der Kolk, who's our product manager and kind of our resident accessibility expert. And he's going to be taking us through the demo in a few minutes. Just quickly, Zoom, I'm sure you're all familiar. If you do have a question, please feel free to post in the chat or raise your hand with the button at the bottom toolbar there. And there will be some Q&A at the end of the presentation and the recording will be available within a day.

(01:00):

So Typefi is, we were founded in 2001, based in Australia and we've got a few different products that are all kind of around helping you do more with your existing resources and helping you publish faster. So we've got our automated publishing software for InDesign, which we'll be showing you in a few minutes. And then we have a few InDesign plug-ins as well. And RunScript, which is cloud services for InDesign Server, let's you run jobs on InDesign Server without the annual licence. And I think I posted some links in the meeting chat with our contact info and some links to those products. But today we're talking about the automated publishing software, which is what you see here on my screen. So this is what's actually doing all this automated layout and composition in InDesign and it can also do this with accessible content and help you produce that accessible content faster. And that is what Guy is going to show us and talk to us about today. So I will now just turn it over to Guy to give you a little bit of background about Typefi.

Guy (02:14):

Thank you Lukas. I'm going to keep the screen, just keep this for a sec, Lukas while I chat before I switch to the other side. So as Lukas said, my name is Guy, I'm been with Typefi for 11 years now, the first couple of years as a solutions consultant, which means I was responsible for implementing workflows at our customers, and the last three years as product owner, product manager, and Typefi Writer is one of my responsibilities as well as accessibility, which is obviously a big, big topic. But at Typefi we try to do everything we can to make sure that accessibility is a core part of what we do because it's becoming more and more important. And as a red-green colourblind person, I cannot tell you how often I cannot properly distinguish good or bad in most business reports because so many business reports use just red and green text to differentiate whether something has gone up or down, which for me especially when that text is in a sea of black, I cannot differentiate those two colours.

(03:34):

So when we're thinking about accessibility, I think a lot of you have seen our previous webinars in this series. If not, I would strongly encourage you to go and watch those recordings. But we always, when it comes to accessibility of content, we usually think about disabilities and then the things that automatically come to mind with that. But a lot of what we do can be very simple things like making sure that you've got a little icon that shows something is going up or something is going down and that icon may be red or green, but at least you've got all your bases covered. Today, you're here to see some of what Typefi does with accessibility specifically. The chart on our screen here is a little bit of an impression of what we do. So Typefi is an automated, a group of software features that allow you to create content auto-magically.

(04:38):

We can see on the item on the screen here. So we've got different kinds of input formats. Most of our customers use either Microsoft Word, which is the demonstration that I'm going to be giving to you today, or XML, one or the other. The other are absolutely options as well, but most of our customers fall in either the Word or XML category where some of that XML may be generated by eXtyles. So those three are our main thing, and even though we support generation of content in up to 30 formats, the most important ones are the ones that you see on the screen on the right. And today I'll be demonstrating generating some accessible content to PDF, HTML, and EPUB, where I am going to have a little caveat, I'm going to be showing you some stuff that's currently under development, so I'm taking a little bit of a risk there.

(05:36):

But as a product team, we've been working a lot on some improvements in that area and I just felt I wanted to share that with you today rather than show you some of the way things went in the past. What you see in the middle is Content XML and you see InDesign sort of the two black circles. And Content XML is our flavour of XML that we use as sort of a wheelbarrow between all the various things on the left: Word, eXtyles, Google, Markdown, all those different kinds of things, and InDesign / EPUB and HTML. So that's a little bit to give you an impression of the overarching design that we do. Go ahead and stop sharing and I'll switch to my screen. I'm just going to go straight into the demo because that's what I like. Going ahead and showing you stuff.

(06:36):

I am aware that there are probably people here that have never seen Typefi, so the presentation balances those aspects of showing you Typefi without driving you absolutely nuts with all the various details. But some of the people watching the video already are aware of what Typefi are able to do but might not be familiar with what we do on the accessibility side of things. So I am trying to cater to everybody. Alright, I'm going to switch to the browser window. So if all goes well in sharing land, you should see a browser window with some things like referring to a marketing SDG sample where SDG stands for the, oh now I forgot. Social, I forgot.

Lukas (07:32):

Sustainable development goals.

Guy (07:34):

Thank you! I've been working on a presentation for school for this for the entire weekend and gosh darn it anyway, yeah, so the sustainable development goals, this is a marketing document that I've been working on for the past couple of weeks in order to be able to demonstrate Typefi and accessibility. And what you can see on screen is the Typefi Server component. Now I'm only showing you this to give you a peek behind the curtain because a lot of you probably, when our current users of Typefi, you probably hardly interact with the Typefi Server at all, but it is to give you a little bit of an impression of what happens in the background. So what you can see on screen is a couple of rocket ships and the rocket ships are the most important ones. So you can see one that's called DOCX to PDF, one that's called DOCX to HTML, and one that's called DOCX to EPUB. So I'm going to focus on the DOCX to PDF first. What the Typefi Server side allows us to do is it allows us to create workflows. So a workflow is a rocket ship and a workflow is comprised of multiple steps. So what you can see here is we start with an Import DOCX action, DOCX being the native format for Microsoft Word. And if you remember I mentioned Content XML or CXML. So this is the first step that...well Lukas is enjoying his breakfast. Sorry, I got distracted.

Lukas (09:06):

Sorry Guy. I wish I could share but it just doesn't work that way.

Guy (09:11):

It's good. Alright, so we start with the Import DOCX action. So the DOCX action takes CXML, takes a DOCX file and it transforms it into Content XML like we saw earlier. We also do some things like copying the fonts to the server and we apply conditions. I'll show you what conditions do in Word in a moment, but conditions allow you, as you can almost imply by the name, you can see that I have a condition for PDF only and HTML only. One of the strengths of Typefi is it allows you to use a single source of content to produce multiple kinds of output as you will see me do in a moment. And using conditions allows you to differentiate content that should only, for example, in this case go into PDF versus content that should only go into HTML and EPUB, which means that you are not managing multiple input files, which a lot of our customers appreciate.

(10:11):

So conditions are an important part of accessibility and obviously they have use cases beyond that like a teacher's manual versus a student book, that kind of thing. But conditions in this case are used to allow us to differentiate content. So I'll show you how that looks in our Word file in a moment. And then we've got our InDesign, Create InDesign Document action. So we're taking that CXML file and we are, I'm actually seeing a mistake. That's why I was, okay, I'm not, I'm seeing a mistake in my workflow, it's not going to impact the presentation and I'm not going to change anything now because changing last minute things breaks things, but that's fine, it won't matter the presentation. So what we're seeing here is our Create InDesign Document action where we take that Content XML and we turn it into an InDesign file. Now you're not going to see that in the way that Lukas demonstrated in his presentation, in his part of the presentation where you saw the pagination being done in InDesign, but imagine that only much faster and on one of our cloud servers.

(11:21):

And then what we do is we export a PDF. At this stage we could export multiple PDFs, so we could create one for print and one for web, but in this case for accessibility, for the demonstration, I'm only showing the web-based PDF or an accessible PDF. And one of the important things after generating that first PDF is in our accessible workflows, we've got an action that allows you to make some important changes to that PDF in order for it to be prepared for the accessibility port. So one of the important things for example, when you're generating a PDF is that the display title must be set to the document title. And that's usually something that has to be done manually in the PDF using a tool like Acrobat Pro, but here we do it automatically. It also allows us to set things like the magnification so that when the PDF is opened, in this case I'm opening it to the page height and it also shows the bookmarks, which is a relatively important part.

(12:29):

So this is an example of a workflow and you can imagine that our DOCX to HTML and DOCX to EPUB actions, workflows, have similar sets of steps in order to prepare them for the generation of PDF or EPUB. So let me, I'm not going to make any changes. If you look at our HTML one for example, it's a lot simpler because there's a lot less steps to it. But we're using that same Import DOCX action, we're using that same condition, but this time it's set to only for content for HTML and then we are generating HTML 5 directly from that Content XML. So this is a little bit of a sneak peek of how workflows work and how they can be set up. Not everybody needs to know every detail around this because like I said, not all of our customers, only a subset of customers, only a subset of people at our customers build workflows.

(13:25):

A lot of people use Writer and generate the content but are not dealing with workflows themselves. Another very important part of our workflow is PDF is important and there's no doubt about that. No matter what these other formats are at this point in our time, PDF is still pretty much the defacto standard. HTML and EPUB are there, but PDF is still what everybody that's interested in producing, it's the first thing on every request that we get put in. So PDF is important and PDF is generated in Typefi in an automated fashion using InDesign and using a template. So what we have here is InDesign and I've opened up the template that I am using for this generation of PDF, so you'll see that in a moment. So one of the things that we pride ourselves on, InDesign is still one of the standard tools for generating good quality, taking into account good typography, that kind of thing.

(14:35):

It is still the standard and therefore a lot of our customers already have a lot of knowledge on how to use InDesign. So building automated templates is obviously something that they would need to learn, but it is not very far from their core set of skills. If you know how to create paragraph styles in InDesign, that already is a very important part of building an automated template. So what you can see here, for example, on the right hand side, you might not be familiar with InDesign at all. I'm not going to go into a deep dive for how to work with InDesign, but I do need for the people in the room that know about InDesign, just give you an impression of what we do and especially what we do from an accessibility perspective. So what you see here on the right hand side are some master pages.

(15:24):

Master pages are the basic designs that allow you to produce the kind of content. So here for example, let me move this out of the way a bit. You can see that we've got a chapter opener master page with an image on the left and room for a text box on the right. We've got a back cover for example that has room for some text. So this is the place where you build the pages that you use in your document. And then we've got some Typefi panels, in this case Typefi sections panels that, in this case Typefi section that allow you to define which master pages are used. So for example here we've got two column with image. So if I double click that, that just tells you which master pages to use, which parent pages, sorry, the name has been changed.

(16:19):

We've also got some elements. So Typefi elements are building blocks. So for example here we've got a cover image or a full page image. Lemme show you this one, it's not spectacular. You can really see that these are building blocks that allow you to take in your content and fill it. So one of the important part when it comes to accessibility is that when you're designing these master pages, you're taking into account some best practises when it comes to accessibility. This is not an InDesign accessibility course, but taking into account the placement of pages or placement of objects on pages so that you get the optimum flow when things go into a PDF and then are read in Acrobat. But probably one of the most important structural components that you're going to deal with is paragraph styles. Paragraph styles, in this case we are making sure that our paragraph styles are defined in such a way that they are, in such a way that they've got the appropriate tags that when this travels through to PDF, if you've seen the previous especially Chad's session, he talked about walking the tags tree and that kind of thing.

(17:44):

Well, it is important to properly define the paragraph styles and in this case one of the things that we do is, in your paragraph styles we set under the export tagging we set the appropriate options. So what you can see here, in this case, I'm only focusing on the PDF aspect, although EPUB and HTML is also set because Typefi used to produce EPUB from InDesign.

(18:15):

So that's why in this particular template I've also set it up, but for now, so here we've set the title to be an H1 tag and you'll not be surprised to note that when I go to my heading one that I've actually set that as an H1 tag. So going through here and making sure that each of my heading styles have this appropriate H tag means that when I export this to PDF, that it is very clear what is or isn't a heading style which makes consumption of my accessible PDF with a screen reader much, much well, not much easier, actually usable. So by setting this up in my template, in my PDF template, we've got that consistency for every PDF that we produce, meaning that everybody benefits from this. We've also got a table of contents in here, which you'll see in the end.

(19:14):

And a table of contents is also, uses those same paragraph styles to generate something that we all benefit from. So that's the InDesign template, just a little bit of a sneak peek behind the scenes on how that works and what we can natively do in terms of accessibility by making sure our options when we're building our InDesign template account for what we want to do with our PDF. Another important aspect, for example, is bullets and numbers. So for making sure that a bullet or a number is actually read as a bullet or a number, it has to be a bulleted or a numbered list in InDesign. But also as you'll see in a moment, in our PDF, because that's the only way it gets tagged properly when it is going through InDesign into our PDF, and if I remember correctly and I edit my bulleted and numbered list, you'll see that I'm using actual bullets in my InDesign template to make sure that when this goes to PDF, that it shows up in the tags tree as an actual bullet and an actual number and so that people using screen readers have that proper identification.

(20:34):

Alright, so that's a little bit the preparation. I'm going to switch to Windows now because the tool that we are using is called Typefi Writer, which is, you're actually seeing the PDF that I've already run previously. It takes a couple of minutes to run, but here is an example of a PDF that I ran through this template just a couple of minutes before this presentation started. So you can see it opened up in that page view that I mentioned, having the single page. Just to double check, can you see the screen with the PDF because I switched to a different one?

Lukas (21:10):

Yep.

Guy (21:11):

Yep. Okay, perfect. Thank you. You can see on the right hand side as I requested in my workflow that the bookmarks be open. Now please note that the bookmarks, just to be very clear, the bookmarks are not the items that are being used by screen readers. That is really the H1 through H6 tags. But this is just to demonstrate that by having structure and by thinking about accessibility, we don't only benefit those of us using a screen reader. Everybody benefits when the structure in your document allows you to have bookmarks like this. I don't need to use a screen reader, but I'm still benefiting from the presence of these beautiful bookmarks on the side because the table of contents is great, it allows me, obviously in this case it's a web-based PDF, so it allows me to go directly to that page, but having that bookmarks open all the time means that I can easily navigate my document.

(22:08):

The fact that this was built based on those heading styles and the table of contents means that somebody using the screen reader is having the same benefits. Everybody benefits by accessibility and that's one of the things that I think was mentioned in pretty much every presentation we've had around this topic so far. Everybody benefits from accessible features. Anyway, so this is the PDF that I generated earlier. I'll show you where I got it from in a moment just because I don't want to pretend that or want you to think that we are cheating somehow, but this is the Word file that this was generated from. So what you can see here is that this document was attached to one of those workflows and one of the most important pieces of structure that we have, because structure is very important in an automated workflow, and it just happens to be that structure is also extremely important in a workflow where accessibility is your top priority.

(23:19):

Anything having to do with structure benefits everybody. So what you can see here is that there is some green tags. So these are Typefi specific tags that have been inserted using Typefi Writer, but you can also see that there's just paragraph styles and in this case I've made it so that you can also see that, for example, sustainable development goals is having title, whereas we've got a paragraph style for body introduction and then body text. And as we scroll through, we start a new section here, we've got our heading one, heading two, our body text and so on and so forth. So paragraph styles are an important structural component and you can see that even in Word, if I start a new document and I apply a title or a heading one to that, it has default characteristics that allow you to see that it is a heading, which is great.

(24:15):

So what do we have here? Well, when it comes to accessibility, obviously your paragraph styles and your heading structure are very important. But another very important aspect is your alternate text. So Typefi has native support for alternate text. If you're seeing, for example here there is a Typefi element called chapter opener image. So I'm going to double click that element to make edits to it, which opens up this dialogue box. And you can see I'm inserting this element from my PC. So if I'm browsing it, depending on how I've set up, I think in this case I haven't set up the previews, but if I set up this to be in, lemme set it up as medium icons. No, there it goes. There, it's presenting the previews. So as I insert this item and if I've got my previews enabled, that will allow me to know what I'm supposed to be inserting.

(25:09):

I'm actually going to cancel this because I don't want to have to reinsert the comment, but you can see that there is an option here to add comments. So in this case, I added the comment, still frame capturing the moment a water... This is not a course about writing alternate text, but having alternate text for your images as has been discussed in pretty much every session we've had around this so far, having alternate text for your images is an important part of an accessible workflow because it allows somebody who is not sighted to still experience this description or what kind of image it is. So we have native support for adding comments to images when you insert them to into a Typefi workflow. And that works. In Microsoft Word we also have native support for, if you are using, this particular workflow doesn't use embedded images because of the kind of design it is.

(26:11):

But if there's an embedded image, I didn't don't know if you know, but let me just give you an example of an embedded image. If I go in here and I do insert pictures and I grab something from this device and I grab one of the pictures from this folder, I'm just going to pick this one. So if you insert an embedded image in Word, which is what this is, there is an option. Oh yeah. So Microsoft Words gives a description and we support this as well. Some of our customers use these kind of embedded images in in Word and if you fill out the Word alt text, we support that as well. Now the interesting thing here is that I'm not here to advocate for alt text or not, and overall I think the alt text generation by Word is questionable. But you can see that my alternate text for this, I'll show you that in a moment when we, I'll find it in the PDF and show you, my description, alt text is also, sorry, not the description, the context is important.

(27:37):

This might be after a natural disaster but the automatic alt text calls it a pile of garbage. So yeah, you definitely want to check your automated AI alt text generation if you use that. Anyway, I'm going to remove this, but like I said, the most important thing is we have native support for alt text. You've still got to write it. We don't have magic tools that allow you to generate alt text auto-magically. That is still a responsibility for your organisation to make sure that you've got somebody that writes alt text. Alright? And like we said here also in Word, it is very important to have native numbered and bulleted lists, especially in our workflow here where we are going to use this Word file and generate an HTML and an EPUB of it.

(28:36):

Yeah, having those native bullets is very important. So here you can see, how can you differentiate a native bullet like a true bullet versus somebody that typed a one followed by a tab? When you are in Word and you click on the numbered or the bulleted lists, you see this sort of grey, I hope you can see it. Let me zoom in a little bit. You can see sort of this, normally you would select it like a character, it doesn't appear, but here it's clearly showing, it selects the number and it shows all the grey for all the items that are part of that numbered list. That's how you know that you're having an actual true numbered list and not something that pretends to be a numbered list. So you need to have true numbered lists and true bulleted lists in our workflows to be able to have them all travel through and be happy all the way from the input to the output. Alright, I'm going to publish this job and then I'm going to take a moment and look at some of the, so what I'm going to do is I'm going to click publish and I'm going to publish that same PDF just because I want you to be able to see what's happening.

(29:53):

Everybody can say that they published the PDF previously. So what I'd like you to observe is that I'm taking this PDF and in Writer I've got the option of choosing. So right now I'm producing the DOCX to PDF because our expectation is that most of the time customers are using the PDF for proofing. So PDF is going to be probably the first thing that gets generated at least for a while, for the foreseeable future. So I've got my pre-flight turned on in case there's anything that's happening like missing images or things like that, but I've made sure that there isn't. So I'm just going to click publish and this is going to take this Word file. It's analysing the workflow, it takes the Word file and it is sending it to that workflow that I showed you earlier. And then it is going to go through the steps in that workflow one by one. And when all of that is done, it is going to give us back a second PDF of the same kind that I already had open when I got in here.

(30:58):

Yeah, so that's what we're going to be doing here. Going to take a quick sip so you can see this gets sent now. Going to mute and take a quick sip of tea because talking always gives a, where's the mute thing? I'll do that in a second. Anyway, so this is running, let me quickly show you on the server, going back to the server where this is happening. So on the Typefi Server we are now seeing our jobs overview. You can see that I ran some jobs earlier and now this PDF is being generated, right? So what I'm going to do is I'm actually going to take the PDF from the job that I ran just before this demonstration started. You saw that and I just want, oh, it's already finished. So on the Word side it's downloading as well, but on the Word side, I don't have Acrobat Pro. So I just want to take the PDF that I generated. So one of the things that I think is important to mention about Typefi is we don't hide anything. So everything that we've generated, including the CXML file that came out of Word, the DOCX file that was originally sent there, but also the generated InDesign file, like the full generated InDesign file, I'm going to download it just because I can and open it up in InDesign.

(32:20):

The full file is there. It's probably going to give me some errors because I haven't prepared for this. So you're not seeing all the images per se, but here is that full generated InDesign file and you can see for example, let me go and find that if I can find it. So everything is here, this is all fully editable. So yes, it's been fully automatically generated, but the InDesign file is there in case you wanted to do something with it, right? So yeah, that's just something that I wanted to share. We don't hide anything. Everything that we generate is right there and available for you to download. I don't think I downloaded the PDF yet. Nope, I forgot to download the pdf. So I'm going to download it and I'm going to open it in Acrobat Pro on my machine so that we can take a little bit of a look at some of the things related to.

(33:16):

So I'm going to download it, I'm just going to download this PDF because that compression in Acrobat takes ages anyway. So here I've got my PDF, I'm going to open that with Adobe Acrobat Pro so that because Adobe Acrobat Pro is a tool that does, allows you, please get away, that allows you to do some accessibility features. So I have to always remember where it is, but there is a...the new interface is still something I have to get used to. There is a, let me see, find text or tool, accessibility check. There we go. Gotta love search. Alright, so I am not going to lie, I don't know. For those of you that were part of the demonstration that Chad gave, I think that was one of the kickoffs of this series of accessibility seminars that we gave.

(34:35):

Chad mentioned that coming from InDesign you can achieve 80 to 100 percent of automation and that really depends on the design and a whole bunch of other things. And within the accessibility community, the accessibility checker for Acrobat is the lowest hanging fruit. There are tools out there that do a lot more than Acrobat's accessibility checker, but as your first steps into accessibility, having the accessibility checker is a good place to start. So let me see if I've made any mistakes in this document and so far it is looking pretty good. It looks like I've made two mistakes with alternate text, but everything else, so the logical reading order is always a question that has to do with, a human needs to look at whether the reading order on the page is correct and the colour contrast is also a manual check, and it seems that I have two missing figures where I forgot to add alternate text.

(35:48):

Let me see which one those are. Oh, okay, that's a mistake on my part and another one, oh, I didn't write alt text for that as well. That's not good. So this allows you to see that there's two places where I forgot to write alt text, but the other ones do have alt text and there's something else that I forgot. Oh, I didn't write alt text for the item on the back cover. So that's something that I could go back into my Word file and I could go ahead and write alt text for those things. But other than that, at least for the Acrobat Pro accessibility checker, this document is pretty darn good. We've got our heading structure are all appropriately nested and so we've got actual headings, we've got actual lists, so there's no failures on that. Our table seems to be pretty good, although we don't have a summary, but we've got actual tables with header rows and all that other good stuff. So except for these alternate text issues, we've got a pretty good accessible PDF here that I wouldn't be too ashamed to send out into the world.

(37:03):

Alright, let's take this that one step further and let's actually generate HTML and EPUB. Now this is the portion where our HTML function. So if you remember I mentioned earlier we used to generate EPUB, for example from InDesign, but we started working on a tool about, I want to say about a year ago where we take that Content XML, we take that structure that we already have in an XML based format, which is so much easier to turn into another XML based format because HTML, which is the core of all of the internet as well as for EPUB, is based on XML. Our EPUB module is, we first worked on the HTML, so that one is very close to completion. Our EPUB module still has some issues. So please, I'm going to generate an EPUB anyway and show you that we can generate all three from this document, but that one still has look through it with how you say through coloured glasses because we're working on that and that still has some work to do. But what I'm going to do, we've proofed our PDF. I could go in here actually, and I could make a change to that back cover. Let me actually do that just because I can. Oh yeah, cover image. I could have sworn I wrote alt text for this, but I must have forgotten.

(38:34):

Well, I did write all text for it. So there must be a different reason why that alt text isn't appearing. Anyway, I am going to publish this document and instead of the PDF that we generated earlier, I am going to now pick one of the other two. Now as you can imagine, this one will be a lot quicker because taking XML and turning it into another kind of XML is a lot quicker than taking XML and taking Content XML and turning it into a fully paginated InDesign file. So I am going to click publish and you can see I picked DOCX to HTML 5 and I'm going to publish that. I'm going to go back to my Typefi Server. I want to be able to see that being done on the fly. And then I want to be able to download my output.

(39:35):

It is thinking about this, alright there it's going. It's sending it off, the job is running, so that means it's going to appear on my Typefi Server any moment. So there we see it running to completion. This should only take a little bit of time. There we go. It's already done. Still blazingly fast. So here what we've done is I've got a zip file. This is our HTML 5 workflow. So I've got a zip file here that I'm going to download that contains the necessary HTML. It's also downloaded on the Word file, but I don't have the tools that I want to be able to show you the background. So I'm going to unpack the zip file. We chose zip simply because that at least allows us to create an output package that contains any of the images that are in this document and that takes any CSS.

(40:33):

So CSS is what makes HTML look good, that kind of thing. So that we can put that all in a package, but obviously this can be customised to whatever you need. So I'm going to double click this and here we've got that same file that we had open here. Let me open up the PDF for reference. Again, we've got that same file that we were looking at in PDF earlier and it has been turned into an HTML file with clickable hyperlinks, a little back to top button, that kind of thing. My skill is not in HTML and CSS. So this is by no means, you can do so much with this, but this is really to show you we can take that same kind of content and turn it into an HTML website that you could then use to upload or modify in whatever way you see fit, including numbered lists that are actual numbered, all those things that we've had before.

(41:36):

You can see here. And I'm going to take a quick moment and show you this in actual code view. I think that's where you can really see where these are the actual tags that you need. So I'm just going to open this really quickly in the actual HTML view. So I'm going to hide. So let me do a quick search on image because that's what, so let me search through all the images. So here you can see the image, the Sustainable Development Goals, and you can see the alt text, which we were looking at in Word earlier is right there in the HTML as well. So single source multiple outputs. Let me quickly go in there and go back to my Word file. I believe by this time you believe me when I say that we're generating that all from the same document, but going in here, going to choose the EPUB even though we still have some work to do on that.

(42:42):

But you're going to at least be able to see that the EPUB is generated. It's in an EPUB wrapper, we can open it up in iBooks and the structure is there. We've got bulleted lists, we don't have the images. That is something that we found wasn't working yet this morning, but you can trust that's going to come very soon the moment that we've worked through those issues. So the job is running again. So I'm going to go back to my Typefi Server here. And as you've gotten used to, ooh, that one ran just as fast. The DOCX to EPUB is also complete. So I'm going to go ahead and download the EPUB from here. And I am going to not open that in my XML editor, but I'm going to open that in the native EPUB viewer. So you can see that the cover is not the same pretty one that we've got for PDF.

(43:39):

You can see that there's some missing images, but still I think the most important thing that I wanted to show here is that we've got that structure, we've got that structure in this case, the title and the heading one, all the headings. My CSS, I wasn't able to get them to be nice and green to match the other ones, but that's just my CSS skills. The content is here. You've got that same structure and the moment that this is ready for full public consumption in a month or two, you're going to be able to take, existing customers of Typefi are going to be able to take their content to the next level with accessibility and customers that might be interested in using or people that are watching this that might be interested in using Typefi to take you to that next level will be able to do. So that's it from me. I'm going to keep my screen on just because there may be questions that require demonstration, but yeah, it's time to open up the floor for questions.

Lukas (44:52):

Cool, thank you Guy, definitely I think I learned something too. Yeah, so if anyone has any questions, feel free to either just raise your hand or post in the chat and yeah, we'll take 'em as they come.

Guy (45:18):

Might have been so clear that there are no questions. That is entirely possible.

Lukas (45:35):

So in theory, Guy, you could create from that Word document for example, that is you've optimised it for these accessible formats with the alt text and things like that, but you could just as easily use that same document to produce other formats as well, like non-accessible formats for print or other things.

Guy (45:59):

Yeah, absolutely. And I think when it comes to accessibility, one of the things is the ability to give your customer options. As with a lot of things, if you put a bunch of accessibility people in the same room, we're bound to disagree on certain things. And because in an ideal world, every document you produce is perfectly accessible all of the time. The reality is that we still have a lot of work to do with each other to make that a reality. But sometimes having a format like a PDF that is designed for a specific purpose, but then having a backup in the form of like an HTML website or having that content in HTML as well, or some of our customers are starting to investigate really publishing to HTML first and making the other options as secondary. Having the options to produce different kinds of content from the same input file I think is huge. I see there's a chat message. I have never heard of NIMAS, so I do not know the answer to that. I can do a really quick by that. I can tell you that by default I don't have support for NIMAS, but let me do a quick Google search. And does it have anything? No, I presume it doesn't have anything to do...

Lukas (47:49):

I got a quick summary here, Guy. It's a technical standard used to prepare digital files to convert instruction manuals into accessible formats.

Guy (48:03):

Awesome, I'd never heard of that. So thank you for enlightening me. Esteban, by default, NIMAS is not one of our 30 native formats, but the benefit of XML is that presuming that there's anything in that format that is related to XML, we can customise it. So I don't know if, that's the best I can give you at this time. So not by default, but the benefits of the workflows, of Typefi workflows and XML in the background, even if that starts from Word, is that potentially it could be customised and if you were interested, we would need to take a look at the format and see in what ways we could do that. So that would be, yeah, thank you for the question, Esteban.

Lukas (49:10):

I'd never heard of that one either, but it sounds like it potentially could be customizable with Typefi.

Guy (49:17):

Awesome.

Lukas (49:19):

Alright, so while we're waiting Guy, can you give an example maybe of a current customer of Typefi and what they're doing with accessibility? Are they just doing accessible PDFs or what type of content are some of our customers using this for?

Guy (49:43):

My first project at Typefi 11 years ago was an accessible workflow for one of our customers in the Geneva area of the world. And they produce multilingual accessible content. So not only are they taking accessibility in mind, they are also producing it in the six United Nations languages. So English, French, Spanish, Chinese, Russian, Arabic, I think off the top of my head. Yeah, so that was the first project I ever did that had anything to do with accessibility. They have been producing accessible PDFs for the past, well, for the past 10 years. It took us a while to get it all set up and they are actually the ones that inspired us to start building the HTML and EPUB export out of, coming from CXML instead of from InDesign because they wanted to be able to do more than just the PDFs.

(50:58):

We also created a customization for them specifically. That's one of the benefits of having this workflow where you've got a Word file and you've got a Word file that is transformed into Content XML as an intermediate format because they did not want to have to type the alt text in the dialogue box that I showed you earlier, which is the native. That's what you get by default with Typefi. But they wanted to be able to write it as regular text. So what we agreed with them is we built a paragraph style, we created a paragraph style called alternate text and the Word document is produced and we take that alternate text that their editors were able to type in each of the languages and we would plug that into the right place in the XML.

(52:01):

So that, sorry I was distracted because I was reading the chat. So in that way, but that's a customization. That's something that they came to us and they said, hey look, we're happy that you can do this natively, but it does take a little bit of time to do that. Is there anything we could do about that? And we said, well this is an option that might work for you. And they agree and they've been using that customised workflow to this day, which I'm very happy about.

(52:34):

So I don't know your first name, but in regards to publication, so there are two things related to your question. One, yes, we can do English and French. I think I saw one of our customers that, on the chat or in the teams, that actually does English and Danish next to each other. So yes, that's absolutely something you can do in InDesign.

(52:58):

If we are then taking on the aspect of accessibility, because you've got to remember that normally one of the things you can do, I don't know if you've observed it in my demonstration, but language is one of the definitions that you can do, that you need to do, that you can do in your PDF. So normally in your properties you also have an overarching language for the entire document, but you can also tag at the paragraph level. So when it comes to accessibility, to be able to differentiate between the two languages, I don't know enough about the tags tree to know if there's something specific you'd need to do manually after the PDF is generated. But I could imagine that a potential solution might be to have two sets of paragraph styles, one for the English and one for the French because the language travels through on that level. But yeah, I don't know enough about the tags tree and languages side by side from a perspective of accessibility, but you could absolutely do that for a PDF, I presume you could do it for HTML too, although I'm not the world's biggest HTML expert, but I could imagine you could create a layout with two divs and define the language there because language is also a definition in several levels of HTML. I hope that answers your question a little bit.

Lukas (54:42):

Yeah, that's an interesting one with the languages on the same page.

Guy (54:47):

Yes it does. When, like I said, when it comes to language, because obviously multiple languages is something that we deal with as well. And for example, one of the ways that you can do this from InDesign, taking a look at the time, yeah, we've got a couple of minutes. You can create character styles that apply specific languages to different locations. So if you've got a French sentence, you can apply a character style that says this is French, which is very important for when your document gets read or when this goes to a screen reader so that it knows to use a different dictionary provided that that dictionary is loaded. So InDesign has that kind of support. Content XML has that kind of support where you can apply languages even at the character level if that's necessary. Alright, not seeing anything in the chat anymore. I think that's it.

Lukas (55:55):

Yeah, looks like it. So thanks everybody for joining us today. If you have any questions, please get in touch with us. We can set up a more in-depth demo for you and go through some more specific questions. But again, thanks everybody for joining and have a great day.

Guy (56:15):

Thank you everybody.