Caleb (00:01):

So I'm Caleb Clauset, VP Product here at Typefi, and as I said just a minute ago, this is sort of a nice midway point for us. This talk that I'm doing right now, this is actually something that I have put together for SSP, which is at the end of the month, and so apologies in advance if I'm going to see you at SSP that if you come to the Industry Breakout session that we're hosting on Wednesday the 28th, that this will be a repeat, but you'll see it here first. So, for 18 years we have partnered with Inera, picking up from where eXtyles leaves off to transform your JATS or BITS or STS XML into beautifully composed publications. But with eXtyles being sunset next year, we're going to step up and introduce Typefi Orion, which is a purpose-built drop in replacement for eXtyles. So the title of my presentation, the sculptor's chisel, the editor's mark, is a little different or unusual, but just bear with me. I think there's some interesting connections between art and language and editorial processes that kind of get into why we're doing what we're doing.

(01:35):

So I love this painting and having just celebrated 20 years now with Typefi, I joined on May 9th, 2005. Crafting these solutions for publishers all over the world, I've definitely developed an appreciation for the gap between representation and meaning. Our publishing systems don't contain actual knowledge. They contain structured representations of knowledge. This distinction is the heart of this painting where Magritte challenges us with this seemingly simple contradiction. It's an image of a pipe with text underneath that says this is not a pipe. And of course he's absolutely correct. It's not a pipe, it's a painting of a pipe. But I love this beautiful imprecision of language and that these gaps between representation and reality is what enables us to convey depth and meaning beyond a simple fact. The way we frame something shapes how it's understood.

(02:47):

Then Banksy, the provocateur, responds to Magritte in 2011 with an actual physical pipe mounted and framed and labelled, "this is a pipe." He even mimicked the cursive of Magritte. His juxtaposition is fun, but it's also profound. It reminds us that context and presentation fundamentally affect meaning. As editors and publishers aren't we engaged in a similar work? We shape understanding through prose and presentation, we decide how knowledge is structured, clarified and delivered. The tools we use for this are not neutral. They influence how we think about and process information. From the conceptual playfulness of Magritte and Banksy, we shift to the halls of the Galleria in Florence, where Michelangelo's sculptures provide meditation on the nature of representation. Here, his prisoners struggle to emerge from their marble confines, caught in various stages of revelation, while the perfection of David towers over us in polished splendour.

(04:01):

The coarse and fine chisel marks in the prisoner Atlas are evidence of his process and echoes of his dialogue with the marble. According to Gombrich in "The Story of Art," Michelangelo saw his role not as creator, but liberator. His envisioned forms were already present within the marble. His role was to chip away the excess to reveal the figure lying dormant within, to reveal what was always present but hidden from view. That chisel isn't a tool of invention, but revelation, gradually exposing the ideal form imprisoned in stone. From the intentionally unfinished prisoners to the perfection of David, Michelangelo didn't just reveal an ideal form but thoughtfully edited it for our benefit. The head and the hands of David are optically exaggerated because the statue was originally supposed to be placed high up on the cathedral of Florence's roof line. The calculated adjustments demonstrate Michelangelo as a sophisticated editor of visual experience. His chisel didn't just free figures from marble. He shaped how we would encounter and understand them, anticipating the relationship between viewer and sculpture across space.

(05:24):

This insight, we shape our tools and thereafter our tools shape us, comes from John Culkin explaining Marshall McLuhan's media theories, and it captures something profound about our relationship with creative tools. Like Churchill's earlier observation about architecture, that we shape our buildings and afterwards our buildings shape us, it recognises the reciprocal influences between creator and creation. Michelangelo's chisel revealed figures from marble, but the properties of the stone itself, its grain, its character, also shaped his artistic choices. He didn't simply impose his will on passive material. He engaged in a dialogue with it, adjusting proportions and form to work with rather than against its nature. This reciprocity exists in all creative tools. We design them for specific purposes, but once created they begin to influence how we think and what we consider possible. The best tools acknowledge this relationship. They extend our capabilities while respecting our intentions.

(06:39):

This reciprocity between tools and thinking takes on a really interesting and fascinating new dimension with AI, but that's a rich topic for its own discussion on a different day. Turning from sculpture to the written word, we find the same reciprocity between tools and thinking in this second or third century papyrus of the Gospel According to John. In this, we can see some of the earliest forms of what we might call punctuation. These mid dots that I've circled originated with Aristophanes of Byzantium, a second century Greek scholar who developed a system of low, mid, and high dots to mark short, intermediate, and long breaths, pauses. They weren't just passive annotations, they actively shaped how the texts were performed and understood.

(07:45):

From these breath marks to inked impressions, punctuation underwent a profound transformation with the advent of movable type. What began as prompts for the creator are now literally cast in lead. Ancient text with minimal punctuation and absent word spaces demanded scholarly attention. Eeaders had to immerse themselves deeply in the material to invest time and intellect. Gutenberg's innovation changed this relationship. His standardised marks accelerated this long evolution towards making text more accessible. This isn't just about the mechanisation of writing, but the process of designing for efficiency of understanding, optimising content for quicker consumption rather than prolonged contemplation. To accommodate Gutenberg, his aesthetic, his standards, and the strict fully justified columns of text, these thin upward angle parallel lines that I've highlighted here are signals of when he's hyphenated a word. So this is a little different way of seeing a hyphenation. But we recognise this as a semicolon and a colon that have developed as a way, again, changing from this idea of a pause or breath into scaffolding meaning within the text, to signify sort of a specific logical relationship between ideas.

(09:18):

What's fascinating here is this sort of denseness of the black letter text didn't just affect the way that punctuation looked. It dictated which marks would work effectively. So instead of a comma here, which might get lost in sort of this forest of heavy vertical strokes, Gutenberg employed the middot again to separate list items. The medium and the message are inseparable. The tools we use determine not only how we express ideas, but which expressions we find possible. The principle of adapting to technological constraints extends to other editorial marks as well. These horizontal marks are conceptually similar to our modern day contractions. They indicate an omission in the text, but while today's contractions are primarily about linguistic efficiency, Gutenberg's omissions were typographic solutions, strategic compressions that enabled his strict columns of justified text. Similarly, this shorthand substitution that appears as a stroked seven, this is again to serve this need to maintain a visual measure in the text. This is a tironian et from around the first century, BCE, and it's the predecessor to our modern ampersand. So that et and the other et that are there, this is a one character substitution for a two character word. Like Michelangelo manipulating Davis's proportions for the benefit of the observer, Gutenberg crafted these editorial marks not merely to represent language, but to guide the reader's eye, shaping how we would encounter and understand text.

(11:10):

I am guessing that you didn't become an editor primarily to wrestle with markup syntax or chase down formatting inconsistencies. The work that we do, what we really care about is helping ideas find their audience with clarity and impact. Which brings me to the core philosophy of why I'm here, why I'm presenting: the tools should free us to focus on meaning not trap us in tedium. This philosophy, that tools should do more, has guided Typefi development for more than two decades. Many of you know us through our InDesign and InDesign Server-based solutions, but regardless of where we operate, we've consistently found ourselves serving as a bridge to modern tools and modern workflows. This role has taken on a new urgency recently and represents a unique opportunity for Typefi. For the past 18 years, we've partnered with Inera and our solutions pick up from where Inera leaves off. That's given us a really deep insight into how eXtyles is used and what is needed as its replacement. So with eXtyles sunset in August of 2026, we're drawing on that experience to launch Orion. Again, this is a purpose built drop in replacement that maintains the workflows you rely on while expanding what's possible.

(12:38):

Today, I'm previewing an early developmental build of Orion that sort of punctuates our progress towards this 1.0 launch at the end of the year. So again, this is why it's a really nice halfway point for us in the year to talk about Orion, what we've done and where we're going. Like the constellation that Orion is named after, it's recognisable and familiar. It is, I said it before, a drop in replacement. Our goal is that you could be working in eXtyles on Friday and continue working in Orion on Monday. So let's get into the fun stuff.

(13:19):

Alright, what you're looking at here in front of you should look pretty familiar. What we have with Orion again is this idea of we want to care about backwards compatibility. And so this is a file that the World Health Organisation has graciously shared with us and allowed us to use in this demo fashion. So when we think about this process that's going through in Orion and applying the structure and cleanup and linking and then eventually into exporting into XML, there's an order, there's a process that we go through. One of the critical processes here is that there's information that is metadata around this particular article that we need to capture and organise an order. So one of the first things that we've been focusing on, and if you've been tracking Orion progress throughout the year, we talked about this earlier, this document information dialogue. And so what we have here is a way to ingest that information if you're again working in eXtyles on the Friday that we can pick up and convert the eXtyles embedded metadata into the Orion format and reflect it and show it to you here.

(14:55):

In addition, that particular dialogue is driven entirely from a web-based interface. And so you as the managing editor or administrator can configure what are all the different characteristics or attributes that you want to capture on a publication by publication basis and even provide in default preset values for certain things. We can choose what kind of option it is, whether it's a checkbox, a text box, or a combo box, and whether it's required, what sequence we want them to do. So you can grab and move these things around. So that what you do here is then reflected immediately within the Orion interface. And so this ability to self-direct, to manage and update and maintain your configuration in your setting is also sort of part of our underlying philosophy in the way that we want our tools to function. That these are tools and you should be able to use those tools for your purposes without having to depend on Typefi to do that configuration for you.

(16:11):

We want to make this as accessible as possible. So once we've set up and captured that metadata, one of the next things that we might do in that early cleanup and preparation of our content is go through again, what should look like a very similar cleanup dialogue to what you have historically with eXtyles. That

you can choose to strip out the extraneous spaces that may precede or end a paragraph, spaces between measurements and say percentage symbols or other sorts of things. Take care of all that upfront to clean up the mechanics of your document. And then again, this paragraph styles, this panel over here, which is distinct from the Word styles pane, again is a very familiar comforting experience from eXtyles. That the same tab structure that you have and the keyboard shortcuts you've created, the way that you've named all this, this is something that we will be able to import directly from your eXtyles implementation and just pull that in directly into Orion.

(17:37):

And again, if you caught this other tab in our web interface, you have the ability to manage and maintain this through this web interface and deploy this automatically to every member within your group. And again, there's an import button so we can import this directly from your eXtyles config. After we've gone through and we've applied the various markup, even the colour scheme that we're using is being imported directly from eXtyles. And so a file that's being worked on in Orion versus a file that was worked on previously with eXtyles will not look different. It's going to look exactly the same to give you that confidence and underlying structure.

(18:25):

Another critical function that we've been working on as of late is the auto-replace functionality. This is the equivalent to the auto-redact engine. Now, one of the things that I want to sort of point out in this auto-replace functionality is that Typefi is built on a modern tech stack. We're built on XSLT. We have a workflow system on the backend that allows you to integrate schematron and other tools into that workflow. And so if you want to use other systems, like I know in our own documentation we've been looking at a tool called Veil, which is a linter. It's like a code correction type tool designed for prose. And so all of this is possible, but we can also again, ingest, import all of your existing auto-redact rules. For some users, this is upwards of 3000 unique regular expressions that we can pull those into Typefi and be able to run them directly through our auto-replace engine. And everything that Typefi is doing, we're trying to be sort of the ideal citizen within the Microsoft Word space. And so nothing about Typefi requires you to flatten your content or to turn off track changes or to accept all the changes or to lose your comments and so forth, that we are going to support all of that as natively as possible and enhance it where we can.

(20:24):

Next up when we jump around within this content, we have references. This is one of those critical things that we want to make sure that is accurate and we know that they're complete. So I can come in here and I'm just going to brutalise this particular one, change some of the data so that I know that it's wrong. And within our advanced processing engine right now, the bibliographic references is leveraging Edifix, which is an Inera product. This is actually built out or derived from the reference cleanup and linking and processing function built into Edifix, but now available as a web service. And so I have two options. I can select individual references and say I want to validate those or if I have no selection, it will automatically detect all of those references and submit those to the Edifix engine and then bring them back and structure them, correct them, link them to both PubMed and Crossref, validate them against the predatory journal database or Retraction Watch and provide comments for you as the editor about any omissions or concerns that you might have there.

(21:58):

This is probably the slowest aspect of the Orion process right now because we're dependent on an external web service. This takes about a minute and a half to go through and validate these 10, nine or 10 references. And it's a blocking task right now because we don't want to change things while this is actually happening in there, but we are now cleaning up and applying the styles and we will shortly replace that and you'll see the beautiful psychedelic colours that we know and love for our references coming into play shortly. And then I should be able to scroll just past number five, I believe. I think it's

the sixth reference that has a caution or notification to the user, but we should also see in the one that I corrected earlier that the information has been filled back in and the number that I changed earlier has been fixed.

(23:03):

We have our PubMed id, we have our DOIs added, we have the beautiful psychedelic colours. All of that is here. And here we have a few annotations from Orion that is flagging some particular references that may need an editorial review to make sure that everything is absolutely correct. Last but not least in this is the all important, being able to export this content into JATS. This is where Orion very tightly ties into the Typefi Cloud infrastructure, and so this will actually be able to watch this happening on the server. So I'll choose export to JATS. This is going to take this Word file, send it up to the Typefi server, and we can now see that this job is running here. Depending on sort of the internet traffic, this can take anywhere from 22 to 50 ish seconds where we're taking that Word file, converting into Content XML.

(24:08):

And from there, wow, this is even faster, 20 seconds, 21 seconds. And so this produces the output into my XML. And so here I have my full JATS XML. If I switch back over to Word, I should also see that same XML file has been opened up inside Word, fully structured, well-formed compliant to the DTD standard. So that is my sort of tight demo of where we are right now in this sort of develop cycle. Now, something you might have noticed that you didn't see is that our advanced processing menu is one item right now. What you're not seeing right now is the citation manager and going through and doing things like the Harvard to Vancouver numbering conversion or name date to number, renumbering of citations, and all of that work that is ongoing. We'll be able to demo that as we get further down the road later this year.

(25:19):

But I wanted to get this sort of end to end example of the progress that we have in front of people as quickly as possible so that you can see that we are serious about this and that we are really focusing on this backwards compatibility aspect and putting a lot of time and effort and care into how we craft this, again, so that it can be as close as a drop in replacement as possible in your workflows because we understand and feel the stress related to a tool that you may have been using for the last 20 years that will be sunsetting in a little over a year from now, August of 2026. And so with that, end of my conclusion, this is why we say do more and that's the presentation on Orion to date. I relish the opportunity to be able to take some questions or chat with you further about where we are, what we're doing and how we can help you do more. So I know there's some chat going on. Let me pull that up.

Guy (26:47):

I would say there's some chat, let's see what is...

Lukas (26:51):

The chat is blowing up!

Caleb (26:54):

Awesome. So let's see.

Guy (26:58):

The first one is from Kathrin around...

Caleb (27:01):

All right, yes. So to answer that question, the answer is it depends. So Orion is going to be available in both a single user Orion Desktop view where everything will happen locally on device. And so all the transform work, all the manipulation is effectively equivalent to the way that eXtyles has operated with you before. With the Orion Workgroup, sort of a larger group, or our Cloud hosted version, that will require some sort of external networking connection, that Orion Workgroup is again designed as an on-premise solution. And so this is going to be deployed within your environment, and so all of your connectivity will be to your instance of the Orion server. If you use the Orion Cloud, then that would require external access. The only other thing to again sort of point out is that the reference correction and linking and validation that's using Edifix regardless of whether it's Orion Desktop, Workgroup or Cloud, that will require an internet connectivity because that is an external web service.

(28:23):

So hopefully that answers that. Luciano, does the paragraph pallette configuration allow for shortcut definition? The answer is yes. That is absolutely something that you'll be able to configure, that you can configure both the, lemme just go over there right now. So within here, you'll be able to configure both the label of how it's displayed to the end user within here, so you can control this label. There's another factor for, okay, when you click Head A, what paragraph style does it apply underneath? There's an abstraction there, and you can also assign a keyboard shortcut to that. That will all be part of this user interface. And again, this is going to be publication based, and so you could have a different set of tabs and styles they would use for different publications. So if you're a service provider that's doing many different journals with different requirements, obviously you're going to have different setups there. So yeah, so hopefully that answers that question. The next question around how will the auto-redact import happen? That's a good question. Guy, do you want to jump on that one?

Guy (29:57):

So right now what we're looking at, and we've been talking about that with Robin for the past couple of weeks since he joined as well. Obviously the import of things like the auto-redact features, the paragraph styles, the existing document metadata configurations, we're building tools for those in the background because it's a one-time process. So if we're looking at, for example, the auto-redact import, we are taking the configuration files because we've been trained by the team on the development and where those files are stored, we will be taking those as a one time and converting them using the tools that are best for that particular kind of content. And then we are placing those, or when we're importing those, it gets transformed into whatever the format of the data storages that we use, whether that's JSON or in the case of the auto-redact, there's a lot of regular expressions that need to be parsed. And then obviously after that we don't, the data from eXtyles is no longer necessary because it's been transformed and uploaded to the Typefi Orion server. So the exact details are, they depend on which kind of conversion, but at a high level, that's what the import button is going to do. It's going to allow us to take the existing configuration files and do magic to them. Engineers are working on that to make sure that they're available and usable for you.

Caleb (31:42):

And to follow up on that, your second question is can you customise this? Can you change those rules? And the answer is absolutely yes. We want, and this is a big part of our job from a documentation perspective, is to provide the information that you need to understand, okay, well here's how I manage and maintain and extend those rules. As I said earlier, a big part of what is behind Orion is saying, okay, we're building in a modern, modern world. We want to make sure that we're leveraging XSLT to the full extent. We want to look around and say, schematron is a useful tool to further validate and apply different business rules that are above and beyond sort of a simple regex compare. We want to provide hooks that enable you to leverage technologies as they come and mature and incorporate that into this process. So

yes, there's nothing proprietary or unique about the way we're doing things. You will have full ability to extend and augment that. Your next question...

Guy (33:08):

Yeah, the next one is about Edifix.

Caleb (33:10):

You do not have to have a separate subscription at all. We take care of all that. It's just a lot simpler if it's just all managed through one, it becomes as transparent and seamless as possible instead of having to maintain, okay, you got to enter your credentials into our system, and then, oh, you ran out of transactions this month. So we have, we're taking care of that for you.

Guy (33:37):

We, because that was an Edifix question, and I see that Solene also had an Edifix question a little later, so I wanted to address that at the same time now that we're talking about Edifix anyway, and then we can switch back to the next one. Solene, the answer to that is yes, and one of the things that we're looking at is when we've got a large amount of references, if we keep the document open, then you are going to be, it's human nature, when something is open, we want to make changes. So one of the things that we are currently discussing is this paradigm of, okay, we don't want you to not be able to use Word while this potentially, I don't think 500 references doesn't seem to be the norm, most of them seem to be less, but we don't want you to not be able to do anything if you do happen to be waiting for a large number of references to be processed.

(34:42):

But we also don't want you to be able to make changes to those files. So one of the things that we've been experimenting with in the background is, okay, we send the document for reference processing, we close the document and you get a popup that says, when your job is done, we open the file for you and say, "Hey, you're now free to continue," which would mean that you would be able to continue working on other files while this processing is happening. So that's kind of as a product team, one of the things that we're working around or trying to figure out, what is the optimum use case? Making sure that you're not able to make changes that could potentially break things later on while also making you feel it's fast by not locking you into a dialogue while it happens to be processing those 500 references. So I hope that somewhat answers your question and anybody else who might've had that question at the same time.

Caleb (35:45):

Yep. You also had a really great question after the Edifix one around, will you be able to configure and create new export XML processes and transformation pipelines? If you're not already familiar with the Typefi server and the workflow engine that is what the Typefi server is that we're leveraging for Orion, this is looking under the hood. This is that when I said export to JATS from within Orion, it ran this specific workflow. And this workflow has five steps that we're manipulating the Word document. We're converting that into XML, we're transforming that with an auto-style function, and then we're exporting that out into the JATS XML. So this is using our standard model of the way we have these style sheets provided in a zip package and so forth, so you can maintain this and you can extend this, you can augment this. There will be a mechanism for you to attach workflows to your environment. And so when you are within the Word side of things that your menu options under advanced processing or your menu options under the export, that you'll be able to extend that as necessary. So if you want to create derivative outputs or you want to run this through a JATS transform and then run a schematron post-process to validate conformance with say the PMC standards, that you'll be able to do that sort of stuff with Orion. This is very much, this is a tool and a tool chain.

(37:57):

I keep hitting these points that your ability to customise this is a bedrock decision in the way that we are developing this. And so the extending and augmenting workflows integrating with other systems that may, similar to Typefi, like, okay, Orion, the output from here is that XML, where does that XML go to next? What other processes does it go to? Does it go to Silverchair? Does it need to go to a review approval process? Are you creating additional outputs from here to create your metadata for submission to Crossref as another channel output? All of that is possible.

Guy (38:53):

Now, I do also want to, everything that we did so far in designing the way that Orion works has all been to make sure that you can extend it. But one of the things that is also true is that we have to have this ready by the end of the year. So in terms of the customization, everything that we've done is to design it to make sure that you can customise it yourself, but it is entirely possible that by the end of this year when the 1.0 version of Orion comes out that we won't be ready yet with all the detailed documentation that the eXtyles team has had about how to do these configurations. So our primary goal if we're looking at the timeline, is obviously to make sure that we've got a 1.0 release ready by the end of this year, and then based on the conversations we've been having with all of you so far, a lot of you, almost all of you want to be ready when that end date of August of next year is there. So when it comes to our resources, our priority is going to be on making sure that we can address, and this is something that we'll get more into over the next couple of months now that Robin has joined us, working out the upgrade plans and that kind of thing. But our primary goal is going to be to make sure that all of you that want to be ready by summer next year that we do the work to make sure that your existing workflows work.

(40:34):

So the customizations absolutely a core part of the design in every decision that we make. Will, by December 1st, all of you be able to do every single thing by yourself yet? Probably not, because the priority is going to be making sure that you're able to, like Caleb said, stop working in eXtyles on Friday and continue working on Monday. So I just want to frame that expectation appropriately. Also, knowing that not all of you want to do customizations. A lot of you probably don't customise eXtyles at all. So I just wanted to address that because I'm seeing that Forsberg C is like, can we control over what sections? Thank you for that. I'm going to make note of that and make sure when we review the feature to make sure that those kinds of things are or not going to be possible or at least add it to the timeline for the future.

Caleb (41:42):

Yeah. One of the things that we have already talked about in that particular area around auto-redact and how to push it further, is looking at how we can leverage things that are already there, but idle. For example, the idea World Health Organisation is a great sort of example for us because we have content that's multilingual, and so being able to identify and restrict certain auto-redact sections based on the XML Lang attribute and say, okay, this, the classic US to UK English transformation. If we've broken it down into parts of words, there's a greater chance that those parts might also appear in other languages. And so being able to say, this substitution only happens within the English language space, that's an extension that's not possible today with eXtyles. But that's a way of thinking about how do we narrow the scope at times, especially because English inherits so many words from other languages that we want to make sure that we do the right thing around that.

(43:25):

And this is where we want. And if you're not already part of the Orion community, this is a great opportunity and another pitch to say, please join. We are hosting these sessions like this and roundtables and one-on-ones to gather your experience and your unique or maybe not so unique requirements and

how we can make sure that that's part of our backlog and part of our thinking process because we very much recognise that there's some things you can change afterwards, you can add on to a system easily. There are other things that may require or be much easier to implement early. And so the sooner we can get that information from you and glean those requirements out, that that can be built into our fundamental base thinking. There was another question that popped up around, do we have plans for an Orion version of eXtyles SI? The answer is a hundred percent yes.

(44:39):

And this is part of why we are putting so much time and effort into the Orion server in these workflows that are running this. Again, that this export to JATS function here is a direct mirror of calling this function on our server. There is a full API that allows you to interact with the Orion server in the same way that you might have already interacted with the Typefi server. That you can call a workflow, you can send a file to it, you can receive that output back and you can tie it into and integrate into other workflow, other systems, other front ends. There's a lot to think about and how this interacts with other systems, but it's something that we are pushing towards. From a timing perspective, it's very likely that the SI version of Orion is going to require a little bit more QA, and so that's not going to be shipping on day one, but that will be following very quickly on in the new year. My goal as the product manager is to get something that's working by the end of this calendar year so that you have eight months in advance to kick the tyres, to do a road ready test of Orion.

(46:19):

If we do our job right, though you can make the decision to adopt Orion very late in your schedule. You don't have to be panicking and worrying at this point that we need to start testing and implementation now to be ready for something in August of next year, that we should be able to carry you through and do more of a "just in time" transition. Alright. Lemme just go through the more questions. So Ron, auto-replace, different rules to different publications. Everything is hanging off the publication level, and so when you think about a publication, you've got document info for that, you've got different paragraph styles for that. You've got different auto-redact rules for that, that everything can hang off that publication. And so there's that I think gives you all the customization and the uniqueness that you need.

Robin (47:32):

Jump in with one observation, Caleb, while we're here.

Caleb (47:34):

Yes, please.

Robin (47:36):

Hello everyone. Nice to see a lot of familiar names in the chat. Just following up on Solene's point about speed and if you have 500 references, and of course one of the things I have brought with me is the imperative that what we don't want to do is make something slower. What we want is something faster. And that's definitely already the ethos within the engineering team. That speed is an important factor here. And while Edifix has a little bit of overhead with the networking time, there are things that are not happening and as in eXtyles, if you remember with documents with lots of tables, there was a whole lot of table processing that had to happen with a lot of these advanced processes—that all goes away. So just to say there are, although you may have that little bit of overhead with the network, what you don't have is all that processing of the tables and the inherits or transformations that are going on within the Orion technology are generally going to be quicker. So I think yeah, certainly something obviously that's front and centre of our minds in terms of looking at how Orion performs that we want it to be quicker, but it's certainly, I think we've already got a massive head start because of the much more modern underlying technology being used.

Caleb (49:08):

Awesome. See your question about the language. I'm just going through the list of, we jumped around a little bit.

Guy (49:17):

I think we've managed to go back. Oh yeah. Luciano is asking in two different ways about the ability to configure and create new export to XML processes and transformation pipelines. That is one question and there was somebody else that had that same question. So the answer to that is again, yes. The way that the export to XML is being designed is so that you can go ahead and make changes. For those of you that are familiar with HTML templating, at its core, it's designed in that way where you have control over, for example, the creation of new paragraph styles. I think that was the question. And adding them to your export mapping, that is absolutely the way that this has been designed so that you're able to do that obviously at first for us to be able to meet your needs for your existing transforms. And then after that, obviously for you to be able to pick that up yourself if you so choose.

Caleb (50:34):

There's a question, a second part in the question from Ron at FASS around the queuing process, which is an interesting question because yeah, for those that are already using the Typefi server for that output to pick up from eXtyles, that we're able to leverage that same infrastructure to provide the workflow engine and the server for the Orion processes. It actually benefits you from a cost perspective because you already have a piece in place. But I think it's key to point out that the job management and queuing system, it's not sort of like one massive queue for all different types of jobs, that if you are running Orion jobs through to produce the XML and so forth, those will occur in parallel to jobs that are using the traditional type of composition engine to make your PDF or make an EPUB. And this is very much something that we are keenly aware of is the scalability and performance and making sure that everything can flow through and that you're not stuck waiting for a job to process. That we want it to be as close to real time. That you trigger this, the job starts processing immediately. So yeah, that's something that we are absolutely working on. But yeah, the idea here is that when you're leveraging this infrastructure that the Typefi server can work in parallel, that many jobs can be happening at the same time. The distinction is that InDesign Server as a component is effectively single threaded. And so that's something that you have to have multiple instances of InDesign Server in order to run multiple jobs parallel. But with Saxon as the underlying engine for XSLT, it doesn't have that sort of single-threaded limitation.

Guy (52:53):

I think that takes care of...

Caleb (52:58):

All right. I think we've hit time effectively, and I think we've hit all the questions. If there's anything else you should know where to find us and we are happy to answer queries, set up one-on-ones, again, get another plug for the Orion community, that is a great way for us to get your feedback and for you to raise the profile of what you want to see in this replacement for eXtyles. It's very much about what can we do to answer the burning problem of eXtyles being sunset, but also how can we plan for improving the process and making this a better solution overall. So we're always very interested in the complaints that you had about eXtyles and how we can make sure that we don't repeat the same mistake or that we provide an improved better experience, whether it's just about the time, that we are more responsive, but also about the functionality and that we can again, do more.

(54:22):

So thank you all for giving me your time, this hour of your days. And if you're going to be at SSP, love to chat further. We'll also be, Lukas, let's see, we've got the I-S-M-T-E in August we'll be at ,and the A-L-P-S-P, did I get that one right? Yeah. In Manchester and Frankfurt Book Fair and Charleston Conference. So we're trying to get out there as much as possible to do the meet and greet, and we'll continue doing these webinars as well to keep you informed of where we are and hope to build that same excitement that we have and what we're doing with you. So thank you again.

Guy (55:17):

Thank you all.

Lukas (55:19):

Thanks guys.