
KATHY SCHNITT: Welcome to the SSAC Evolution of the Resolution Work Party teleconference on Thursday, 13th of October, 2022. Barry, I'll hand it back over to you.

BARRY LEIBA: Thanks. We'll start off by thanking Jeff for being the only one who actually made some comments on the text that we asked people to make comments on. And I guess what we're going to do is start off talking about that text. We start a bit with what Jeff's comments were, but I think, Andrew and I decided to mostly agree with what was there, and he's just making comments. So Andrew, why don't you go ahead and move on? Take it away.

ANDREW MCCONACHIE: Jeff, you made two comments. I think you were kind of in this first one, you were basically agreeing with the text or did you want more text added here? And if you're speaking, I can't hear you.

UNKNOWN SPEAKER: Jeff, you're off mute, but there's no sound coming from you. Are you double muted?

JEFFREY BEDSER: That's because I'm using the wrong device.

Note: The following is the output resulting from transcribing an audio file into a word/text document. Although the transcription is largely accurate, in some cases may be incomplete or inaccurate due to inaudible passages and grammatical corrections. It is posted as an aid to the original audio file, but should not be treated as an authoritative record.

ANDREW MCCONACHIE: There you go. Now you sound like the resonant Jeff, that we're used to hear.

JEFFREY BEDSER: So this whole thing about users discovering resources on the web. And the whole issue was in my head. Domain names were never about discovery, they're about reference. And search was actually a novel thing called discovery. But underneath it was the DNS, which was reference.

And the sentence that I kind of tweaked to, was that sentence that goes as early as 2005 kind of goes the wrong way. It sort of says, well, DNS was search, and then search was search. And the answer really was search engines fill the gap that didn't exist. And they didn't replace domain names. But they replaced the DNS as the primary means of referencing resources because search not only did discovery in research, but it also did indirect reference. I sent you some Google search terms, you discover them.

RUSS HOUSLEY: I think there's a step missing, though, which is that the web created that need and before that, anonymous FTP, nobody created a spider to walk anonymous FTP sites although, maybe in hindsight, they could have. But the thing that web was created to do was publish and so then there was a reason to search.

BARRY LEIBA: I see the point that Jeff is -- I think Jeff is referring to it as reference is follows the point that we were trying to make, the idea is that from the user's point of view, and this section is about the user's perspective, from the user's point of view, searching has replaced typing in names. But maybe reference is really the right word there.

ANDREW MCCONACHIE: Or access.

BARRY LEIBA: Or access, yeah.

JEFFREY BEDSER: Because it's not replacing it. And all it did was suppressed domain names down one layer into being invisible, but they didn't go away and they didn't get replaced. Search engines don't send you to an IP address.

UNKNOWN SPEAKER: So I think access resources on the web is better than reference, because I don't think that people generally say go to Google and type in www.tiktok.com. They just open Chrome and type in Tiktok.

UNKNOWN SPEAKER: That's because they go to Bing not Google. What's the matter with you?

BARRY LEIBA: But yeah, I think in that particular sentence, if you just change discover to access, I think that reads exactly the way we intended.

UNKNOWN SPEAKER: And I know it's not only that it's that they have improved in quality and ease of use. It's that there is now only one place to enter a thingy. And so even if you want to access www.microsoft.com, you type it in the same spot as you type in a search thingy, generally.

BARRY LEIBA: So there's an interesting change in the user interface where the earlier browsers had an address bar and a search bar. And now we've merged them and the browser takes care of figuring out what it's going to do with type there. I don't know if we want to mention that somewhere in here or not. But Andrew, take note of that, and think about that.

ANDREW MCCONACHIE: That's mentioned.

JEFFREY BEDSER: I think it's like two paragraphs down. This trend has continued.

ANDREW MCCONACHIE: I knew I had that in the earlier text that was here. And I moved that down to 6.2, which I haven't yet changed to be for the perspective of software developers. But I'm wondering if I don't mention it here that we have this kind of magic bar at the top of browsers, I should.

BARRY LEIBA: And there was a QR code mentioned later on as well. So we do keep getting into the different ways we've morphed this.

ANDREW MCCONACHIE: The user share. So it really be here. So here, I should mention the, what is that called?

BARRY LEIBA: Omnibox.

ANDREW MCCONACHIE: Omnibox.

JEFFREY BEDSER: Generally, or at least that's one organization's name.

UNKNOWN SPEAKER: Does everybody call it that?

JEFFREY BEDSER: I think most do. But it's...

UNKNOWN SPEAKER: A term I've heard used is magic box.

ANDREW MCCONACHIE: Magic box. I'll figure out the right term to use.

JEFFREY BEDSER: The issue was, as far as I can see that there were two input things at the user interface level, one was precise and pedantic and was intolerant of any imprecision, which is the DNS and the URL. The other one was the search term, which was sort of tight how you feel typos, spelling mistakes, who cares? And the issue was I suppose, that the URL bar was just a nightmare. Everyone got shit wrong. Whereas the search engine thing, kind of auto corrected for the human. And so it was almost logical that combining it and making it fuzzy, was considered to be a step forward in the UI. The DNS was always there. But this was kind of taking human fuzzy input. And trying to resolve that into something quite precise the DNS could deal with.

ANDREW MCCONACHIE: Think the text says mostly what you just said. Is there anything that we need to add to address what you just said?

JEFFREY BEDSER: I suppose not really, it depends on how you sort of think about when you read that sentence and that area.

ANDREW MCCONACHIE: The last sentence of this paragraph?

JEFFREY BEDSER: I'm going to read quote John Linson, I suppose because he always said the DNS was never a search engine, it was never imprecise. There was always precision. That's what the DNS is. And search engines actually performed a different function. And then we say they're easier to use. They're more forgiving of user mistakes, and they're more adaptable, absolutely. But the issue was the DNS was never discovery. It was intended to be discovery, it was never built for discovery.

UNKNOWN SPEAKER: We just have that sense before.

UNKNOWN SPEAKER: The DNS was never intended to do discovery.

ANDREW MCCONACHIE: Well, then how come there's this book from 2005 called the Signposts in Cyberspace?

JEFFREY BEDSER: Written by whom?

ANDREW MCCONACHIE: National Research Council.

JEFFREY BEDSER: Or probably people can be wrong.

ANDREW MCCONACHIE: It's not actually a sucky book.

UNKNOWN SPEAKER: They're also using a metaphor and metaphors change over time.

UNKNOWN SPEAKER: But it is a signpost., it's over there.

UNKNOWN SPEAKER: It's not really a signpost as much as it's an address on a shingle, its called like that, a shingle.

ANDREW MCCONACHIE: But shingle names point you to other parts of the DNS.

JEFFREY BEDSER: And there's NAPTR records. But in all fairness, the DNS was not designed as a search thingy.

UNKNOWN SPEAKER: Never was thought it to be. If you misspell Starbucks, when you type starbucks.com, you're not going to get where you need to be. If you type that into Google. Well, if you type that into Google, or Bing, or

whatever, Google will helpfully tell you searching for Starbucks, if you really want to go to Starbucks, type here.

JEFFREY BEDSER: Is looking how many see names there are to Starbucks in terms of typos. But we could get sidetracked into corner cases.

UNKNOWN SPEAKER: We always get sidetracked into corner cases.

BARRY LEIBA: I think I can add some text to this paragraph to address Jeff's concern. Well, not just Jeff's concern, I think the concern we're coming up with here, which is that the DNS was not designed as a search discovery mechanism. And that there's a big difference between precision required by the DNS or URL input, versus a search engine. So I'll add a couple sentences on that. And I'm going to skip down here to where we're discussing QR codes. And you had another comment, Jeff about a QR code resolving to a URL. And that's a good point, because I didn't actually say that in this text. So I changed this last sentence so that, the audience reading this can understand it, a QR code and may resolve to URL, and it's just another layer of indirection.

JEFFREY BEDSER: What else does it resolve to?

BARRY LEIBA: They can do a lot of different things.

UNKNOWN SPEAKER: When you use Wi Fi, DPP, it resolves to a public key.

JEFFREY BEDSER: I didn't know that. Well, maybe I've never been taken to a QR code that actually had a public key.

ANDREW MCCONACHIE: My QR code resolve to populating my apple wallet with my vaccination card. You can put all sorts of random crap in the Mirchi.

JEFFREY BEDSER: Fine, no. It's just bits.

UNKNOWN SPEAKER: It encodes texts or images or whatever.

JEFFREY BEDSER: In the same way that base 64 does.

ANDREW MCCONACHIE: The bottom line is that from the user's point of view, it's another way to go access something on the web, generally.

JEFFREY BEDSER: But with COVID, everybody got used to that being a way to go get the menu for the restaurant they walked in and stuff and so it just is a squishing of the user interface into a push a button and you don't even see the URL, it opens a PDF. And you never even see it on your screen.

UNKNOWN SPEAKER: Correct, it's magic.

JEFFREY BEDSER: It's not magic, but it's completely hidden.

UNKNOWN SPEAKER: To most users; I think it's magic.

JEFFREY BEDSER: It's expected though.

BARRY LEIBA: Actually, I suppose the true magic from the user's point of view would be that you walk in the restaurant and when you look at your phone, you see the menu and you didn't have to do anything. That would be truly magic from the user's point of view.

JEFFREY BEDSER: That's my point was to get back to this. Most of the text is there, apart from the fact that QR codes, basically a level of indirection to adapt to the user. For most cases, and certainly in the context of this document,

where we're talking about the end user perspective, the QR code is basically an alternate to the precision of the DNS. And it kind of hides it inside an implicit context. Where did you see that QR code? And so it's just indirection rather than a new invention.

BARRY LEIBA: I agree though.

UNKNOWN SPEAKER: Hey, Barry, I think I know how to do that magic.

BARRY LEIBA: The one that I said about when you walk in, it's there?

UNKNOWN SPEAKER: We'll talk about it over a beer in London.

BARRY LEIBA: Excellent.

JEFFREY BEDSER: Meanwhile, I'll get a bus ticket, hit off the page, see you.

BARRY LEIBA: I'm going to add something here about indirection.

JEFFREY BEDSER: Well, you should say, in redirection that is often not even shown to the user.

BARRY LEIBA: And that shows up in a lot of ways sometimes, if you have an app installed, the QR code will just directly from your point of view, just open the app. And you cannot even see the browser anymore.

ANDREW MCCONACHIE: Those are the only two comments on the text.

JEFFREY BEDSER: No, 2.1 go back up. 2.1 examples.

ANDREW MCCONACHIE: Oh, right, this one?

JEFFREY BEDSER: The kind of the document, not view, the document kind of has some definitions. And then says examples with these rather weird headings domain name and DNS day, actually doesn't explain itself.

ANDREW MCCONACHIE: But that table with the thought that it might be helpful as we were talking about stuff. And we just never got to talking about that. We spent one call talking about it a little bit. But we haven't gotten back to

it. And so that's why it's still there. We're just trying to focus on Section 6.

JEFFREY BEDSER:

Fine. But that was the only other comment I had, I couldn't put it into context.

BARRY LEIBA:

I can write an intro for 2.1. I guess the larger question, are those the right three columns? String domain name and DNS name? Hold on a little bit?

UNKNOWN SPEAKER:

I think so.

UNKNOWN SPEAKER:

The only question would be whether you wanted to do notations annotations to say why this thing doesn't fit, maybe? I know we said it above. I just don't know what to say in places where you said this is for the quasi-phone number that is the domain name, for example, then followed by one that isn't. Is it going to be immediately evident as to why those two are X'ed-out or even if that's a footnote. This doesn't qualify because of this, if we're using this as an illustration, just a thought.

UNKNOWN SPEAKER: And for that one in particular, there's 212-555-1212, it's a domain name, it's a DNS name, it's a telephone. It all depends on the context you're looking at the end. And is going to be an important point in other parts of the discussion.

JEFFREY BEDSER: We will add the 161.arpa for that same phone number.

UNKNOWN SPEAKER: No.

UNKNOWN SPEAKER: What we might want to do.

UNKNOWN SPEAKER: It might make people hurt.

BARRY LEIBA: What we might want to do is add in some like a separate column or something or a bunch of footnotes, saying things like this is all of these things. And as with many it is context dependent or something like that. I don't quite know how long words are.

JEFFREY BEDSER: I see your point. You add a Comments column and we could, when it's both say why or something,

BARRY LEIBA: Something like that. My issue is or a concern is the notes, having a table with the notes field as its own entire paper gets weird and tricky. So maybe they shouldn't be like asterisks or something or call outs or something.

UNKNOWN SPEAKER: They could refer to where, for example, that's defined, you could say, fails requirements. And then because we're already listing out what those requirements are, fails Bla. And then you don't say the whole thing. But look at where we've listed that out.

UNKNOWN SPEAKER: It's funny that I keep trying to scroll down on your son's screen, which doesn't. But the one that I was, you might make a note to add India.ra to that sort of unlisted examples where it's a domain name, it's a DNS name, it's also a human's name. That's what she goes by. And again, it's context dependent, what it means to you right now.

UNKNOWN SPEAKER: Like that.

UNKNOWN SPEAKER: Arid, India.arid. You can look her up at some point, she's a singer.

UNKNOWN SPEAKER: That is a domain name, DNS name. But it is also, to me a meaningless string or an opaque string. So we need something else to put in here. This is also somebody, a person or something, because otherwise it just they chose some random set of strings.

UNKNOWN SPEAKER: That's where the comment field. Would you need to comment field?

JEFFREY BEDSER: I don't know what we put in the comment field for like, why example.com, is a domain name and a DNS name? Just because they are.

UNKNOWN SPEAKER: Does the person really use the .in their name?

UNKNOWN SPEAKER: Oh, that's true.

UNKNOWN SPEAKER: And to warn, we don't need to have a comment in every law.

ANDREW MCCONACHIE: So for example, here, we would say domain names and DNS names can't start with a plus sign.

JEFFREY BEDSER: But I think we also need to say above, this is also a phone number or something. Maybe we need a box thing for and that kind of is going to be checked almost on all of them, context dependent or something.

ANDREW MCCONACHIE: I think that's going to go in the in the intro or the outro to the table. So yes, in some of the comments, we might know other things it also is and then either before or after the table explained what we're getting at here about the context dependency on this stuff.

BARRY LEIBA: So we'll have like a catch all statement at the beginning that these are largely context dependent, so we don't have to talk about it individually for each row.

ANDREW MCCONACHIE: The only thing we might want to talk about individually is what another common context for this string might be. Like phone number for example. If we want to leave those question marks there, which I think is fine. This would be the place for an asterisk where you could really dive into why `_printer.example.com` is tricky.

BARRY LEIBA: But I actually didn't want us to spend a lot of time on the table today. I think we will get back to that at some point. So I'd like to go back to Section 6. And first class, do we think at least we're now done with 6.1? Is 6.1 say the kind of thing we want to say? And do we want to start

looking at the topic of 6.2, which is the view from software developers' point of view?

JEFFREY BEDSER:

I have one more thing. And I'm about to put it in as a comment, if I can figure out how to insert a comment here.

See if I can do this. Sorry, just playing with this silly Google interface. Why can't I comment? Here we go. I'm trying to sort of sort out why we have this section. And what I've added in as a comment is the issue that we started to talk about context. And we are so used to, as in this phone call communicating in a realm of implied context. So when I use the term in the context of this call, most of you actually grok what I'm all about without me having to explain it in my new detail. And the problem is the DNS doesn't play that game. There is no implied context, the realm of the DNS is global, and everything is precision. And as we tried to lift this technology into general human use, the problem is that humans didn't naturally adapt to what the DNS did. And so we actually had to introduce ambiguity and collision into those interfaces to actually match what humans do. Because our natural desire was to make this technology useful, rather than make humans adapt to the technology. And so I was trying to motivate most of the rest of section 6.1 in going, there is nothing wrong with this. And it is perfectly natural that it happened. And it's never going to stop. Never going to stop. Indeed, and I think that's a large part of why we're doing this. Because we want to look at what those

other contexts are, and how names on the internet are, increasingly, the increasingly need different contexts to understand them.

JEFFREY BEDSER: So I just thought that introductory text right at the front, would actually motivate the body of the analysis as to particular mechanisms of indirection and ambiguity by saying, this is why we did this.

BARRY LEIBA: I think that would work really well in six. So before we go into any of the perspectives, we have that context, like right here. That works.

UNKNOWN SPEAKER: In that motivates why it is, and we're talking about different perspectives entirely, really. because everything is so contextual.

ANDREW MCCONACHIE: And then as we move into the perspective, from the software developer, we get to where the user doesn't really need to understand the context in a fundamental way, the user generally just knows what the context is because of what the user is doing. But the software developer needs to understand the context in order to do the right thing with the string the user gave it.

JEFFREY BEDSER: And potentially resolve ambiguities to the extent possible. The whole idea that, you present everything as a question to the user, the

certificate is a dud, do you still want to go there is kind of the useless dialogue. And the software developer is actually trying to answer all of those questions automatically. In other words, supply context.

ANDREW MCCONACHIE: I always say we have to get out of the business of asking users questions that they're unqualified to answer.

BARRY LEIBA: Any other comments on 6.1?

JEFFREY BEDSER: There's got to be a corollary for that comment, though Barry. Lawyers never ask a question they don't already know the answer to, at least good lawyers down. At that magic bar seems to provide context, or guess at it. Guess at it based on what other people have asked by guests,

BARRY LEIBA: Or what they know about you, because they have more data about you than you do about yourself.

UNKNOWN SPEAKER: That might be something we might go into here as we talk about context.

JEFFREY BEDSER: Well, it's the issue of the software developer, because by planting to the search engine, the software developer is saying, "I have no idea of the context of the use of my application, either. I'm going to let search engines." This default system, language, locale, time, gender, user, apply the default context. Other software developers are saying, "I know my realm, I know my realm very well. I'm writing a Tor application, whatever. And I'm not going to do that. I'm actually going to answer the boxes, tick the ambiguities, make deliberate choices, because this is my intention." I'm still working in a realm of DNS, and I'm still working in a realm of user ambiguity. But my software is making particular choices that I don't want to punt on.

ANDREW MCCONACHIE: Sure, there's going to be some software that's going to be very rigid about it and wants to be, and other software that wants to be more flexible and adaptable. And others still others that call out to somebody else's software to handle this problem. And that's software may take into account the user's location, the user's previous behavior, information that has been gathered about the user and so on. Sure.

JEFFREY BEDSER: I agree completely.

UNKNOWN SPEAKER: With these attributes, specific identifiers, like why it's nicer to have a Facebook identifier or a Twitter identifier, because it knows things about you.

JEFFREY BEDSER: I thought that was a reason not to.

BARRY LEIBA: That is the other side of that.

JEFFREY BEDSER: But the issue for the software developers to actually understand the realm of choices available, and to deliberately make them assume nothing or answer every ambiguity, make deliberate choices. And there's no single answer for software developers, none at all. But it's their role if they want their software to be usable, to actually resolve most of these points of ambiguity one way or another, to actually make the software, conform to their desired behavior.

BARRY LEIBA: That's good to hear that. This section here, which is titled software developers is the old text from 6.1, which I haven't yet put into this. I think when I first wrote 6.1, I was too nerdy. So that's why I pushed us down here into software developers, but I think it still needs to be massaged. And I think I need more of that intro, what you're talking about Jeff, the real solutions that software developers are trying to provide and why it is they make the kind of choices they do.

JEFFREY BEDSER: Yes, and stress that there's no right answer and no wrong answer because particular applications and use contexts, require various assumptions which are not general, they're just for that particular application or service.

UNKNOWN SPEAKER: So if we can really talk about this section anymore, it sounds like we're done with 6.1. So I don't know if there's, was there anything else you wanted to talk about today, Barry?

BARRY LEIBA: Unless we just want to riff on the software developers stuff and give you folder for later editing, no.

UNKNOWN SPEAKER: I'm cool with folder. More fun or more better.

UNKNOWN SPEAKER: Russ, is there anything specific you want to chat about?

RUSS HOUSLEY: No, that seems fine. I was thinking, we could always go spend another 100 hours on the definitions. But the software development is a fine place to go next.

UNKNOWN SPEAKER: So if there's any more chatter we can do about the perspective of software developers that will help Andrew flesh this text out better, let's have it.

BARRY LEIBA: And we're talking about software developers that have to use domain names where we are talking about software developers that use internet identifiers?

JEFFREY BEDSER: I think it's the latter because the last part of the previous section is the context. And so, I think trying to say here, how does that context get folded into software?

ANDREW MCCONACHIE: And I think that you want to say almost definitely, the default name resolution context for most applications is blah, blah, blah for the beginning.

BARRY LEIBA: That's true.

ANDREW MCCONACHIE: I should say, internet applications, because, when you get into.

BARRY LEIBA: That's probably true.

NANDREW MCCONACHIE: Names and applications. It's not about internet names. Could just be about variable names.

BARRY LEIBA: I think also, we should probably, like the whole advertisers rely on bit seems kind of randomly in, feels kind of weird. And that's at the end of that page. That's kind of just like, how, what are not software developers or anything like that?

ANDREW MCCONACHIE: That's leftover from the user perspective.

BARRY LEIBA: And I wonder, actually, if this is -- when we're talking about applications here, traffic does anything that's actually wrong or not applicable. But when we're talking about this is primarily desktop applications. What we're seeing is true for mobile applications as well. But a huge chunk of mobile applications don't so much expose the idea. I'm looking at my phone as we speak. And a number of them don't really use the internet. Not as many as I would like when I happen to be on an aero plane and nothing works. But they are often the framework that the developer interacts with when writing for example, an iOS application is not the sort of standard desktop application thing of take URL, find name, extract name, resolve name, with many of the more modern frameworks, it's more Fitch resource, and you handle a URL. You don't deal with the DNS; you deal with the concept of a resource or a URL.

And the fact that there is stuff under that is kind of a pick to you. I'm not sure if I'm managing to word that well.

ANDREW MCCONACHIE: You mean it's opaque to the developer because they're just calling, some system call an iOS which handles a URL for them, and they don't have to ever make a call to like libc to resolve a DNS name.

BARRY LEIBA: Yeah. So if one has a look and just trying to find like...

ANDREW MCCONACHIE: So they're basically high level API's that...

BARRY LEIBA: There's so much...

ANDREW MCCONACHIE: [Crosstalk 00:40:10] developers.

BARRY LEIBA: Trying to find there must be a good example that I can just point out. But the standard examples of talking to a REST API these days is something like this, where you go put in the chat. Requests URL was that, request, stop, get blah, blah, blah. And so, if you ever see or, almost, if you ever see anybody doing a DNS lookup from a Python app, probably something has gone wrong. So technically, yes, that URL does

get stripped apart by the requests library. And it resolves the name but the user, the application developer views increasingly, the thing as a URL, not a protocol://a domain name/a path, it's a thing or a resource that you get.

ANDREW MCCONACHIE: Sure. And then I'm sure there's many other different types of identifiers besides just URLs, if you're developing on the Facebook platform, I'm sure there's some kind of Facebook identifier. And on the Facebook API like references a user, which actually resolves to a URL, which then resolves the domain name. So, there are many different kinds of layers and different types of abstractions that developers have to deal with. So they don't even see names is what we're really talking about here.

BARRY LEIBA: If you have a look at like, the most used programming languages as of 2022, it goes Python, Java, JavaScript. JavaScript has 9% of the market share of programming languages. And in JavaScript, it is, for all intents and purposes, impossible to do a DNS lookup. Actually, the only real way you can do it is you do it through a request, basically. And so JavaScript programmers, don't deal with domain names.

ANDREW MCCONACHIE: They make request to URLs.

JEFFREY BEDSER: Or I'll actually say they make requests about identifiers. And do not themselves, explicitly customize the context or environment where that request comes. And it's up to the rest of the environment. So if I'm using resolvable as DNS, and relying on server push out there, for some reason, or somehow, from their apps point of view, it's still just an identifier. And whether the global DNS is indirectly invoked, or some server has pushed that identifier and its resolution at me beforehand, I don't care. I'm trying to write abstracted code at the top level. And I don't really want to define how the bits go at the lower level. It's not my job.

ANDREW MCCONACHIE: Well, the browser won't let you for security reasons.

JEFFREY BEDSER: So the other part of this, what's probably missing in this section, is this whole issue of generality, abstraction and libraries. No one writes, what I would call first level code, go down every rabbit hole, write code for everything all the time, import this massive collection of libraries. Write a three line orchestration program, you're done.

ANDREW MCCONACHIE: And to some extent, that's a modern invention, or well, not invention, but that's a modern trend, because now we have all these really robust libraries and API's out there. In the early 1980s, we didn't.

JEFFREY BEDSER: The libraries are out there be everything has a cost and no one wants to pay it. So reuse, reuse, reuse.

ANDREW MCCONACHIE: And computers are really fast. Who cares if the library and like wasting all these [inaudible - 00:45:18]. Well, I've got, I definitely have some good ideas for how I'm going to restructure this section. And I don't know if I need to take up people's time anymore.

BARRY LEIBA: Well, I'm happy to call it a call. And to say we'll meet again next week. So, next week, we should have some text in 6.2, that actually reflects what we want to say here and we can discuss that.

ANDREW MCCONACHIE: Sounds good.

JEFFREY BEDSER: I have an apology for next week. I'm flying off to [inaudible - 00:46:17], maybe five.

UNKNOWN SPEAKER: You're not going to NANOG, what's wrong with you? Good gracious me. Why would you possibly, fair enough. Is there anyone here who is going to be a non-author I can chat with.

UNKNOWN SPEAKER: Warren will be drinking alone. That's a scary thought.

UNKNOWN SPEAKER: Warren doesn't drink.

UNKNOWN SPEAKER: So next week...

UNKNOWN SPEAKER: Next week we will...

UNKNOWN SPEAKER: I think Barry...

UNKNOWN SPEAKER: Timing coordinates the week after that.

JEFFREY BEDSER: Will be the workshop, right?

UNKNOWN SPEAKER: Yes. And the week after that will be the workshop. So we will not meet, and then we'll be able to meet after that?

UNKNOWN SPEAKER: No, we are meeting at the workshop.

JEFFREY BEDSER: There is a scheduled session at the workshop.

UNKNOWN SPEAKER: I didn't remember that we had a slot. That's good. Sorry, I think I'm losing.

JEFFREY BEDSER: You're losing what?

UNKNOWN SPEAKER: I'm losing network quality anyway, so it's a good time for me to drop off.

JEFFREY BEDSER: All right, be well, everyone.

UNKNOWN SPEAKER: All right. So we'll see everyone next week.

UNKNOWN SPEAKER: Andrew, do you have a minute?

ANDREW MCCONACHIE: Who was that? That was...

RUSS HOUSLEY: That's Russ.

ANDREW MCCONACHIE: I'll see you.

RUSS HOUSLEY: It's not long, and it's a little thing, that probably would be a perfect fix for you to cause to happen. And I was looking at SSAC 121 when I went to it, from the SSAC documents page, it comes up in the PDF obviously, tab. And in the tab or window identifier associated with that window that's open, it starts off with the words draft. So whoever needs to go in and erase draft, so that it no longer safe, not the document itself, just the tabs.

ANDREW MCCONACHIE: This is metadata inside the PDF?

UNKNOWN SPEAKER: Is there something?

UNKNOWN SPEAKER: They do that?

ANDREW MCCONACHIE: I'll strip the media data. I'm supposed to strip the metadata out of the PDFs, but sometimes I forget.

RUSS HOUSLEY: So many things happened behind the scenes just like we were talking about. And the only other question I have, it's really in a way more of a question, I guess for writing joy. There was talk about putting in to do a presentation on SSAC 121 for the ICANN.

ANDREW MCCONACHIE: The ICANN do this symposium. Geoff Houston is going to do it. I could send you the presentation. But basically, it's the exact same one that you gave at ICANN 75.

RUSS HOUSLEY: And that's just fine. I hadn't heard anything. And usually I'm the one that gets asked to do those.

ANDREW MCCONACHIE: I should have told you Russ. It just came together. And then we looked at the slides, they were like these slides are fine. So I stripped all of the ICANN branding from the slides. Other than that, they're the exact same slides that you gave at ICANN 75.

RUSS HOUSLEY: And is Geoff actually going to be physically there?

ANDREW MCCONACHIE: Yes.

RUSS HOUSLEY: That's a very good thing.

ANDREW MCCONACHIE: I will be physically there as well.

RUSS HOUSLEY: So I'll probably just do my remote thing. And watch, good.

ANDREW MCCONACHIE: You don't have to present this time. You've presented it enough.

RUSS HOUSLEY: I did, since I kind of expected to be asked I hadn't heard anything. That's why I figured it was a good thing for me to ask what was happening so, how can it be? That sound great?

ANDREW MCCONACHIE: Anything else?

RUSS HOUSLEY: No. That's all I had.

UNKNOWN SPEAKER: Hang on and watch modern work. Warren is just having side chats with Kathy. Bye all.

ANDREW MCCONACHIE: Can I close the call?

UNKNOWN SPEAKER: I think that Andrew can close the call.

UNKNOWN SPEAKER: He's gone.

UNKNOWN SPEAKER: Bye.

[END OF TRANSCRIPTION]