

Document Number: P3575R0

Date: 2025-1-13

Authors: Michael Wong

Project: Programming Language C++, SG14 Games Dev/Low Latency/Financial Trading/Banking/Simulation/Embedded

Reply to: Michael Wong <fraggamuffin@gmail.com>

## **SG14: Low Latency/Games/Embedded/Financial Trading virtual Meeting Minutes 2024/11/13**

### Contents

**Minutes for 2024/11/13 SG14 Conference Call**

**1**

### **Minutes for 2024/11/13 SG14 Conference Call**

Chaired by and Notes by Andre Kostur. Thank you.

OK, meeting completed: Really only got through the 1 paper. The other two had no comment, partly because Patrice couldn't make it to present.

TL;DR: Discussion on P3429 went for a while. Jens had much to say, and we need to forward the comments on to Jonathan (author). The other two papers had no presenter. Arthur raised a concern that allocator-aware inplace\_vector won't make it to 26 and thus will essentially be dead.

Minutes:

> Agenda:

>

> 1. Opening and introduction

>

> ISO Code of Conduct

> <

>

<https://isotc.iso.org/livelink/livelink?func=ll&objId=20882226&objAction=Open&nexturl=%2Flivelink%2Flivelink%3Ffunc%3Dll%26objId%3D20158641%26objAction%3Dbrowse%26viewType%3D1>

> \*>\*

>

> ISO patent policy.

>

[https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common\\_Policy.htm?nodeid=6344764&vernum=-2](https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2)

>

> IEC Code of Conduct:

>

> <https://www.iec.ch/basecamp/iec-code-conduct-technical-work>

>

> WG21 Code of Conduct:

>

> <https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures>

>

> 1.1 Roll call of participants

Andre Kostur, Lauri Vasama, Paul Bendixen, Gianluca Delfino, Jake Fevold, Michael Wong (to start us off), Arthur O'Dwyer, Jonas Persson, Khalil Estell, Matthew Butler, John McFarlane, Jens Maurer

> 1.2 Adopt agenda

>

> 1.3 Approve minutes from the previous meeting, and approve publishing

> previously approved minutes to ISOCPP.org

Arthur was asking about where to find previous minutes.

Michel: They are sent to the reflector and as a paper.

>

> 1.4 Action items from previous meetings

>

> 2. Main issues (125 min)

>

> 2.1 General logistics

>

> 2024 planning

> C++23 and C++26 status

> CPPCON SG14/19

>

> Future and past meeting plans

>

> \* Jan 10, 2024 02:00 PM ET: Games DONE

> \* Feb 14, 2024 02:00 PM ET: Embedded DONE

> \* Mar 13, 2024 02:00 PM ET: Cancelled due to Tokyo 3-18-23

> \* Apr 10, 2024 02:00 PM ET: Finance DONE

> \* May 8, 2024 02:00 PM ET: Games DONE

> \* June 12, 2024 02:00 PM ET: Embedded; St.louis 6-24-29 DONE

> \* July 10, 2024 02:00 PM ET: Finance DONE

> \* Aug 14, 2024 02:00 PM ET: Games DONE

> \* Sep 11, 2024 02:00 PM ET: CPPCON Sept 15-20 so cancelled DONE

> \* Oct 9, 2024 02:00 PM ET: Embedded

> \* Nov 13, 2024 02:00 PM ET: Cancelled Wroclaw F2F

> \* Dec 11, 2024 02:00 PM ET: Finance

>

> 2.2 Paper reviews

> Embedded:

\* P3429 <meta> should minimize standard library dependencies

Chair: Note this was raised in the email list yesterday.

Paul: I'm not an author. We'd probably want this document, but try to avoid using facilities that aren't available in the embedded/freestanding environment.

Paul: Note the motivating is both for the unavailable facilities, but also just for compilation time.

Paul: Since gaming was specifically mentioned, we should solicit input from the gaming community/representatives.

Paul: Noting the document talking about using more basic types instead of the heavier-weight ranges, string\_view, et al

Paul: Particularly calling out Poll 2 (std::meta::info\_array) and how it removes the dependency on std::vector

Andre: Why `char*` instead of `std::array`

Paul: Calling out to `source_location` as an existing practice.

Suspects there may be resistance by depending on further libraries.

But could be a good point to raise.

John Mc: `std::array` could increase compile times.

Paul: It may pull in other things, such as exceptions.

Jens: Since the author isn't here, what's the involvement of this group, and how do these comments get to the author.

Paul: We're doing it here because we were concerned about the reflection paper's direction w.r.t. freestanding constraints. I think this would be prudent to discuss. Perhaps we can email him the notes?

Andre: Isn't this being brought up in Wroclaw

Paul: Yes, the idea is to talk about it here and bring it to Wroclaw.

Jens: Do we have a channel? Who is sending the author this minutes, or is there some other channel being used?

John: Mc: We should.

Lauri: `const char *` over `std::array` because we don't know the size.

Jens: Many items. `std::array` would result in different sizes and thus different types. Note that the paper number has digits swapped.

Section 6: didn't see the rationale why `<string_view>` and `<vector>` should go.

Making reflection-range exposition-only. Probably a good idea: we don't need a human-spellable concept if humans aren't expected to directly use it.

Section 7: we should add a name to things which can have the range-based for loop run on it.

Section 8: We may want a modifiable container here. Should `info_array` take a template parameter?

Isn't one of the design concerns for reflection is interaction with the std library?

Section 9: Don't think this will work as it's hard to initialize an `initializer_list` onself.



Section 10: We might be able to generalize this to some sort of range.

Section 11: If this is a consteval type, there can be no runtime values, and thus no ABI concerns.

Khalil: Perhaps only `std::vector` is really an issue. `std::span` doesn't seem to be a big problem. Speaking from an embedded perspective, I'm not worried about these dependencies.

John McF: `std::vector` isn't accepted in gaming environments, and compile-times are a concern.

Khalil: Reflection itself will probably take some compile-time on its own.

John McF: Anticipating using reflection for serialization. Currently done through some sort of annotation.

Khalil: Seems like in the compile-time comparisons, `ranges` seems to be a big offender. Perhaps we should have separate comparisons of `ranges`, `vector`, and `string_view` separately.

Jens: `Ranges` is certainly large. Since all of them are templated, the

compiler has to pay for parsing them.

Arthur: Agree with Khalil, somewhat push back on Jens: ranges pulls in `_lots_`. `libc++` is working to reduce the amount that ranges must pull in. But that work seems to be implementor-specific, and not easily accessible to end users. For example: forward-declaring just `std::vector`. I don't understand why we're including ranges? Since we have `reflection-range`, I suppose that will pull in ranges.

Jens: Maybe the authors aren't familiar that by showing `#include <ranges>`, then end-users will be able to rely on `<meta>` pulling in `<ranges>`.

Arthur: Supporting Khalil that perhaps the benchmarking is capturing too much and that users won't be necessarily including ranges if they include `meta`.

Paul: Supporting Khalil. `vector` seems to be the greatest issue. I do see that at some point we'll want a mutable container. Thanks for the input.

> \* P3132 Accept attributes with user-defined prefixes

Chair: Author not present

> \* P3134 Attribute [[asserts\_rvo]]

Chair: Author not present

Arthur: There's a thread about allocator-aware inplace\_vector. And this won't be discussed in Wroclaw, and thus will miss C++26.

Andre: Out altogether, or for C++29?

Arthur: This is a bugfix to something going into C++26, so it either needs to go into 26, or it's not feasibly happening.

Khalil: Was discussing with other people in St Louis about something sounding like this. Could that be taken up for 29?

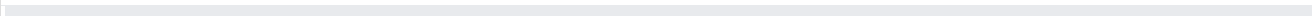
Arthur: Perhaps you were thinking of "small vector"? We could come up with some new thing, but wouldn't be able to fix inplace\_vector.

Khalil: Could you give a quick synopsis as to why to add an allocator awareness?

Arthur: the allocator is responsible for a number of types, and does the allocator propagate? It's responsible for construction and destruction of objects. We need a way to get an allocator into the elements of the inplace\_vector for their use.

Khalil: I see the value. How much support do you need to get this back on 26's radar?

Arthur: This is a political issue, we'd need to find a good champion for this.



>

> SG14/SG19 features/issues/defects:

>

>

[https://docs.google.com/spreadsheets/d/1JnUJBO72QVURttkKr7gn0\\_WjP--P0vAne8JBfzbRiy0/edit#gid=0](https://docs.google.com/spreadsheets/d/1JnUJBO72QVURttkKr7gn0_WjP--P0vAne8JBfzbRiy0/edit#gid=0)

>

> 2.3 Domain-specific discussions

>

> 2.3.1 SIG chairs

>

> - Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin

> Holmes, John McFarlane

>

> - Financial/Trading chairs: Robin Rowe, Staffan Tjernström

> Carl Cooke, Neal Horlock,

> - Games chairs: Rene Riviera, Guy Davidson and Paul Hampson, Patrice Roy

>

> - Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson

>

> 2.4 Other Papers and proposals

>

> 2.5 Future F2F meetings:

>

> 2.6 future C++ Standard meetings:

> <https://isocpp.org/std/meetings-and-participation/upcoming-meetings>

>

> -

>

> 3. Any other business

> Reflector

> <https://lists.isocpp.org/mailman/listinfo.cgi/sg14>

> As well as look through papers marked "SG14" in recent standards committee

> paper mailings:

> <http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/>

> <http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/>

>

> Code and proposal Staging area

> <https://github.com/WG21-SG14/SG14>

> 4. Review

>

> 4.1 Review and approve resolutions and issues [e.g., changes to SG's

> working draft]

>

> 4.2 Review action items (5 min)

>

> 5. Closing process

>

> 5.1 Establish next agenda

>

> 5.2 Future meeting

>

>

> \* Jan 10, 2024 02:00 PM ET: Games DONE

> \* Feb 14, 2024 02:00 PM ET: Embedded DONE

> \* Mar 13, 2024 02:00 PM ET: Cancelled due to Tokyo 3-18-23

> \* Apr 10, 2024 02:00 PM ET: Finance DONE

> \* May 8, 2024 02:00 PM ET: Games DONE

> \* June 12, 2024 02:00 PM ET: Embedded; St.louis 6-24-29 DONE

> \* July 10, 2024 02:00 PM ET: Finance

> \* Aug 14, 2024 02:00 PM ET: Games

> \* Sep 11, 2024 02:00 PM ET: CPPCON Sept 15-20 so cancelled

> \* Oct 9, 2024 02:00 PM ET: Embedded

> \* Nov 13, 2024 02:00 PM ET: Wroclaw F2F

> \* Dec 11, 2024 02:00 PM ET: Finance

On Wed, Nov 13, 2024 at 6:10 AM Andre Kostur <[andre@kostur.net](mailto:andre@kostur.net)> wrote:

>

> Morning (for me): 2FA is kicking in, and needs confirmation from your device.

>



> On Wed, Nov 13, 2024 at 5:07 AM Michael Wong <[fraggamuffin@gmail.com](mailto:fraggamuffin@gmail.com)> wrote:

> >

> > Hi Andre, Thank you for volunteering to chair.

> > I have set the zoom link to username/password as:

> > [fraggamuffin@gmail.com](mailto:fraggamuffin@gmail.com)

> > Sg14&sg19?!

> >

> > Please check to see if that works. I might be up at 3 am my time, but more likely not. Its likely you will have to take notes. I have set the agenda.

> >

> > On Tue, Nov 12, 2024 at 10:08 AM Andre Kostur via SG14 <[sg14@lists.isocpp.org](mailto:sg14@lists.isocpp.org)> wrote:

> >>

> >> I intended to be at tomorrow's meeting (and virtually in Wroclaw), if you'd like I could chair.

> >>

> >> On Mon, Nov 11, 2024 at 9:54 AM Michael Wong via SG14 <[sg14@lists.isocpp.org](mailto:sg14@lists.isocpp.org)> wrote:

> >>>

> >>> Thanks Paul for the call out.

> >>> Given that we are not having a meeting in Poland, I really would like to have this call on Wednesday at 2 pm ET.

> >>>

> >>> I will unfortunately be travelling. I am still trying to figure out if I can manage the call from the opposite timezone around 2 am. Even if I were to get a volunteer to chair in my absence, I would still need to start the Zoom call which can only be done from my account.

> >>>

> >>> But first thing first, any volunteer chair? Thank you. Regards

> >>>

> >>>

> >>> On Mon, Nov 11, 2024 at 11:37 AM Paul M. Bendixen via SG14 <[sg14@lists.isocpp.org](mailto:sg14@lists.isocpp.org)> wrote:

> >>>>

> >>>> Hi

> >>>> I'm not entirely sure if the idea was to have the meeting on Wednesday, but if we are, I'd like to add a paper to discuss.

> >>>>

> >>>> P3429 "<meta> should minimize library dependencies" seems like the most important addition to the reflection discussion seen from the embedded side.

> >>>> Even if <vector> and friends get in freestanding somehow (seems Ben is working on this) it could delay adoption in the embedded space if something like this paper is not considered.

> >>>> Since the games industry is explicitly called out in the paper, it might be of interest there as well.

> >>>>

> >>>> The paper is on the agenda for Poland, so perhaps a quick discussion beforehand could be good?

> >>>>

> >>>> Best regards

> >>>> Paul M. B. Bendixen

> >>>>

> >>>> --

> >>>> . - - ./. -/.. -/.. - . ./- . . ././- ./- . ././- . . -/./- ./.. - - . ./- - ./- -/.. -/.. ./.. - . . -/.. . -  
./- - -/.. -//

> >>>> \_\_\_\_\_

> >>>> SG14 mailing list

> >>>> [SG14@lists.isocpp.org](mailto:SG14@lists.isocpp.org)

> >>>> <https://lists.isocpp.org/mailman/listinfo.cgi/sg14>

> >>>

> >>> \_\_\_\_\_

> >>> SG14 mailing list

> >>> [SG14@lists.isocpp.org](mailto:SG14@lists.isocpp.org)

> >>> <https://lists.isocpp.org/mailman/listinfo.cgi/sg14>

> >>

> >> \_\_\_\_\_

> >> SG14 mailing list

> >> [SG14@lists.isocpp.org](mailto:SG14@lists.isocpp.org)

> >> <https://lists.isocpp.org/mailman/listinfo.cgi/sg14>