

Document Number: P2683R0  
Date: 2022-10-15  
Authors: Michael Wong  
Project: Programming Language C++, SG14 Games Dev/Low Latency/Financial  
Trading/Banking/Simulation/Embedded  
Reply to: Michael Wong <michael@codeplay.com>

## SG14: Low Latency/Games Dev/Embedded/Financial Trading/Banking/Simulation Meeting Minutes 2022/02/09- 2022/10/12

### Contents

Minutes for 2022/02/09 SG14 Conference Call .....	1
Minutes for 2022/03/09 SG14 Conference Call .....	7
Minutes for 2022/04/13 SG14 Conference Call .....	14
Minutes for 2022/06/08 SG14 Conference Call .....	22
Minutes for 2022/08/10 SG14 Conference Call .....	31
Minutes for 2022/10/12 SG14 Conference Call .....	36

### Minutes for 2022/02/09 SG14 Conference Call

Short call today. Thanks all for replying, no worries.

On Wed, Feb 9, 2022 at 9:13 AM Michael Wong <fraggamuffin\_at\_[hidden]> wrote:

> Topic: SG14 Low Latency Monthly This meeting is focused on Games. We can  
> discuss Patrice's WIP games paper. Are there any other topics, and sorry  
> for the late notice.  
>  
>  
>  
> Hi,  
>

- > Michael Wong is inviting you to a scheduled Zoom meeting.
- >
- > Topic: SG14 monthly
- > Time: 2nd Wednesdays 02:00 PM Eastern Time 1900 UTC (US and Canada)
- > Every month on the Second Wed,
- >
- > Join from PC, Mac, Linux, iOS or Android:
- > <https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJdT1loeXpKbTcydz0>
- >
- > Password: 789626
- >
- > Or iPhone one-tap :
- > US: +12532158782,,93151864365# or +13017158592,,93151864365#
- > Or Telephone:
- > Dial(for higher quality, dial a number based on your current location):
- > US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1
- > 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833
- > or 877 853 5247 (Toll Free)
- > Meeting ID: 931 5186 4365
- > Password: 789626
- > International numbers available: <https://iso.zoom.us/u/abRrVivZoD>
- >
- > Or Skype for Business (Lync):
- > <https://iso.zoom.us/skype/93151864365>
- >
- > 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%3DII%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

>  
> 1.2 Adopt agenda

>  
> 1.3 Approve minutes from previous meeting, and approve publishing  
> previously approved minutes to ISOCPP.org

>  
> 1.4 Action items from previous meetings

>  
> 2. Main issues (125 min)

>  
> 2.1 General logistics

>  
> Future meeting plans

>  
> F2F vs hybrid:

>  
virtual is viable for SG, people can only be limited to 1 or 2 day  
attendance

core groups: need an entire week, is a loss in throughput if they dont meet  
F2F, 5 days 6-7 hours a day, hard to do that even with weekly meeting  
super telecons to address major designs, but its still hard  
F2F need hybrid support? money? facility?

hard to attend F2F as observer, need connection/proxy

hybrid is preferred if at all possible  
F2F all in same timezone, 12 hours opposite  
subchairs adjusted their schedule

- >
- > \*Feb 9, 2022 02:00 PM ET/1900 UTC: Games\*
- > \*Mar 9, 2022 02:00 PM ET/1900 UTC: Embedded\*
- >
- > 2.2 Paper reviews
- >
- > Discussion on Low Latency/Finance topics
- > P2300
- > Swift
- >
- >
- >
- > Discussion about Games topics:
- >
- > P2388R1 - Minimum Contract Support: either Ignore or Check\_and\_abort
- > <<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>>
- >
- > Patrice's WIP on Games issues.
- >
- > Finance topics from July 14, 2021.
- >
- > <https://lists.isocpp.org/sg14/2021/06/0636.php>
- >
- > <https://lists.isocpp.org/sg14/2021/07/0642.php>
- >
- > 2.2.1 any other proposal for reviews?
- >
- > Deterministic Exception for Embedded by James Renwick
- >
- > [https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low\\_cost\\_deterministic\\_C](https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C)

[exceptions for embedded systems.pdf](#)

- >
- > Freestanding?
- >
- > 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: 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
- >
- >
- > \*Feb 9, 2022 02:00 PM ET/1900 UTC: Games\*
- > \*Mar 9, 2022 02:00 PM ET/1900 UTC: Embedded\*
- >

## Minutes for 2022/03/09 SG14 Conference Call

On Tue, Mar 8, 2022 at 10:49 PM Michael Wong <fraggamuffin\_at\_[hidden]> wrote:

> Topic: SG14 Low Latency Monthly This meeting is focused on Embedded.

>

>

>

> Hi,

>

> Michael Wong is inviting you to a scheduled Zoom meeting.

>

> Topic: SG14 monthly

> Time: 2nd Wednesdays 02:00 PM Eastern Time 1900 UTC (US and Canada)

> Every month on the Second Wed,

>

> Join from PC, Mac, Linux, iOS or Android:

> <https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJmT1loEXpKbTcydz0>

o

> Password: 789626

>

> Or iPhone one-tap :

> US: +12532158782,,93151864365# or +13017158592,,93151864365#

> Or Telephone:

> Dial(for higher quality, dial a number based on your current location):

> US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1

> 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833

> or 877 853 5247 (Toll Free)

> Meeting ID: 931 5186 4365

> Password: 789626

> International numbers available: <https://iso.zoom.us/u/abRrVivZoD>

>

> Or Skype for Business (Lync):

> <https://iso.zoom.us/skype/93151864365>

>

> 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%3DII%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
- >

Paul Bendixen, Andreas Weis, Brett Searles, Detlef Vollmann, Henry Miller, John McFarlane, Ka Ming Chan, Matthew Butler, Ronen Friedman, Michael Wong, Rene Riviera, Ben Saks, Andrew Lumsdaine

- >
  - >
  - > 1.2 Adopt agenda
  - >
  - > SG13 events handling for graphics
- event handling is low latency related, interrupt handler,



- > 1.3 Approve minutes from previous meeting, and approve publishing
- > previously approved minutes to ISOCPP.org
- >
- > 1.4 Action items from previous meetings
- >
- > 2. Main issues (125 min)
- >
- > 2.1 General logistics
- >
- > Future meeting plans
- >
- > \*Mar 9, 2022 02:00 PM ET/1900 UTC: Embedded\*
- > \*Apr 13, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- > \*May 11, 2022 02:00 PM ET/1800 UTC: Games
- > \*June 8, 2022 02:00 PM ET/1800 UTC: Embedded
- > \*Jul 13, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- > \*Aug 10, 2022 02:00 PM ET/1800 UTC: Games
- > \*Sep 12, 2022 02:00 PM ET/1800 UTC: Embedded
- > \*Oct 12, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- >

CPPCON:Sep 12

Embo++ is virtual and is free and can have GatherTown: Mar 25/26  
free tickets may run out

- >
- >
- > 2.2 Paper reviews
- >

- >
- > Discussion on Embedded:
- >

> Review latest mailings:

> P2532 Removing exception\_ptr from the receivers concept

>

design change now that executor is not in

if sender use exceptions, the receiver must handle exceptions, but if

noexcept sender then receiver has none

So for embedded if we don't have exceptions, then this would be maskable interrupts so it won't be handled

is it UDP? sender/receiver is much lower level

More redesigns likely for executors, should we have an SG for executors, but currently it was in SG1

sender/receiver supports events

Based on the last meeting and the discussions here.

> P2544 C++ Exceptions are becoming more and more problematic

>

multi core systems with locks in the exception mechanism has severe penalties

need global lock on the unwinding table so as to be safe for when you load dynamic library even though it is rare you need the, unless you use transactional memory which could be an ABI break even if we use advance Lock free mechanisms

it's the auxiliary data structure that the unwind mechanisms

can we put these tables in data segment instead of code segment?

changing the eh mechanism requires substantial knowledge to get it right James Renwick and Bjarne should have a look at this

We might want to chime in here.

> /Paul

> P. S. P2327 de-deprecating volatile received a "consensus" straw poll.

>

> started from SG14 and was accepted

P2327 straw poll consensus as a defect report to go in for 23, only the bitwise operator for volatile and people were surprised it is not atomic original paper was still good because volatile is not to be used on multiple threads, but for hardware register arithmetic

> Discussion on Low Latency/Finance topics

> P2300

> Swift

>

>

>

> Discussion about Games topics:

>

> P2388R1 - Minimum Contract Support: either Ignore or Check\_and\_abort

> <<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>>

>

> Patrice's WIP on Games issues.

>

> Finance topics from July 14, 2021.

>

> <https://lists.isocpp.org/sg14/2021/06/0636.php>

>

> <https://lists.isocpp.org/sg14/2021/07/0642.php>

>

> 2.2.1 any other proposal for reviews?

>

> Deterministic Exception for Embedded by James Renwick

>

> [https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low\\_cost\\_deterministic\\_C\\_exceptions\\_for\\_embedded\\_systems.pdf](https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf)

>

- > Freestanding?
- >
- > 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: 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
- >
- > \*Mar 9, 2022 02:00 PM ET/1900 UTC: Embedded\*
- > \*Apr 13, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- > \*May 11, 2022 02:00 PM ET/1800 UTC: Games
- > \*June 8, 2022 02:00 PM ET/1800 UTC: Embedded
- > \*Jul 13, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- > \*Aug 10, 2022 02:00 PM ET/1800 UTC: Games
- > \*Sep 12, 2022 02:00 PM ET/1800 UTC: Embedded
- > \*Oct 12, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- >

## Minutes for 2022/04/13 SG14 Conference Call

On Tue, Apr 12, 2022 at 10:02 PM Michael Wong <fraggamuffin\_at\_[hidden]> wrote:

- > Topic: SG14 Low Latency Monthly This meeting is focused on Low Latency/
- > Finance.
- >
- >
- > Hi,
- >
- > Michael Wong is inviting you to a scheduled Zoom meeting.
- >
- > Topic: SG14 monthly
- > Time: 2nd Wednesdays 02:00 PM Eastern Time 1800 UTC (US and Canada)
- > Every month on the Second Wed,
- >
- > Join from PC, Mac, Linux, iOS or Android:
- > <https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJuT1loeXpKbTcydz09>
- > Password: 789626
- >
- > Or iPhone one-tap :
- > US: +12532158782,,93151864365# or +13017158592,,93151864365#
- > Or Telephone:
- > Dial(for higher quality, dial a number based on your current location):
- > US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1
- > 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833
- > or 877 853 5247 (Toll Free)
- > Meeting ID: 931 5186 4365
- > Password: 789626
- > International numbers available: <https://iso.zoom.us/u/abRrVivZoD>
- >
- > Or Skype for Business (Lync):
- > <https://iso.zoom.us/skype/93151864365>
- >
- > 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
- >
- Staffan Tjernstom, Andrew Lumsdaine, Ben Saks, Guy Davidson, Henry Mille, Jens Maurer, Patrice Roy, Rene Riviera, Ronen Friedman, Michael Wong, Timur Doumler
  
- >
- >
- > 1.2 Adopt agenda
- >
- > 1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org
- >
- > 1.4 Action items from previous meetings
- >
- > 2. Main issues (125 min)
- >
- > 2.1 General logistics
- >
- > Future meeting plans
- >
- > \*Apr 13, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- > \*May 11, 2022 02:00 PM ET/1800 UTC: Games
- > \*June 8, 2022 02:00 PM ET/1800 UTC: Embedded
- > \*Jul 13, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- > \*Aug 10, 2022 02:00 PM ET/1800 UTC: Games
- > \*Sep 12, 2022 02:00 PM ET/1800 UTC: Embedded
- > \*Oct 12, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- >

> 2.2 Paper reviews

> Discussion on Embedded:

> Review latest mailings:

> P2532 Removing exception\_ptr from the receivers concept

> Based on the last meeting and the discussions here.

> P2544 C++ Exceptions are becoming more and more problematic

> We might want to chime in here.

> /Paul

> P. S. P2327 de-deprecating volatile received a "consensus" straw poll.

>

>

> Discussion on Low Latency/Finance topics

>

>

> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p1839r4.pdf>

>

presented by Timur

hard to access byte of object, not type punning which affects aliasing

char can alias any other type, not undefined.

char -> int, value does not fit in a char, can chop pointer, which can be undefined behaviour,

lots of old code that does reinterpret cast to print the bytes, and this was valid until C++17,

proposal is to un-UB this

what about launder? it is covered in a section

7.4.1 p3 can only access current active member of union

what happens if it is a null ptr? null\_ptr\_t with be all 0 on most platforms,

P0137R1 made it into UB, and we think that was just an inadvertant mistake while fixing the lifetime

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2016/p0137r1.html>

How is this affected/different when using std::bit\_cast instead of reinterpret\_cast - if at all? bit\_cast is different way, just a wrapper on memcpy, here we want to look at the byte as they are in memory. Cool - would be good to clarify that in the paper - as far as I can tell so far, every bit\_cast implementation - when used on pointers at least - seems to just effectively do no memcpy and just is equivalent to reinterpret\_cast.

ptr to vtable, 4 byte offset issue from Andrew

bit cast gives u fully live object

memcpy can only copy between object of same type

no requirement in C++ that random object need to have continuous value



representation, so should not work for virtual functions or virtual base classes - but that is not what the wording said in p6.1 has a requirement for object contiguous property - then that could be a bug in the wording may need to limit the effect to object that is well behaved - contiguous? looking at 7.1.2 green blob of words

I can confirm that classes with virtual functions are not standard-layout classes.

We have a trait named `has_unique_representations` that comes close to expressing contiguousness in terms of (essentially) having no padding, but it's a Library trait :

[https://en.cppreference.com/w/cpp/types/has\\_unique\\_object\\_representations](https://en.cppreference.com/w/cpp/types/has_unique_object_representations)  
In the sense of "I'm unsure if it can be used in a Core language discussion"

follow on paper:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2020/p1945r0.pdf>  
has definition of contiguous bytes of storage, so please add a cross-reference

The reference.. <https://eel.is/c++draft/intro.object#8.4>

> Patrice's paper on Making C++ better for game developers.

>

V4

How to proceed with this big paper?

do we agree with general principles

- simple teachable C++

- no negative performance impact

- debugging matters (fast debug builds) that can be turned off in release mode

(due to high turn over)

finance care about how the code will run

ranges, monadic types, futures, a lot more processing embedded in C++ Std

makes it hard for RTOS, deep call stack of ranges is hard to debug

Traits section removed

Compile time computing requested by games

allow overloading by `constexpr` arguments

use case is string literal, and runtime produced by character pointer, so

can we see use case that don't involved string literals

Source location with `nameof` (C# style) for logging core dump

Is it compiler or runtime info?

data members should not be variable unless they are static data members,  
may be object scope

also having reflection support would be nice

reflection features should ask SG7, enum inspections

Not going to keep all of this, but remove some and give form to the rest of  
the request

Compile-time string interpolation

For the runtime a nicer looking syntax that defies the lexing convention of  
C++ is novelty

Determinism: need to know it never allocates from embedded side, with  
static assert to say that might be ok

also important for audio (but static assert could be error prone)

and safety critical applications like autonomous vehicles

Options:

Or take option one by one

take self evident ones first

removed solved ones

identify missing pieces

.

>

On Hive from Matthew Bentley

I have feedback to share from the final LEWG talk (until next month) - will  
take some time when you're all ready - shouldn't take more than 5 minutes

;) )

Priority parameter has been removed, is biggest news

Probably right thing to do

>

- > P2300
- > Swift
- >
- >
- >
- > Discussion about Games topics:
- >
- > P2388R1 - Minimum Contract Support: either Ignore or Check\_and\_abort
- > <<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>>
- >
- > Patrice's WIP on Games issues.
- >
- > Finance topics from July 14, 2021.
- >
- > <https://lists.isocpp.org/sg14/2021/06/0636.php>
- >
- > <https://lists.isocpp.org/sg14/2021/07/0642.php>
- >
- > 2.2.1 any other proposal for reviews?
- >
- > Deterministic Exception for Embedded by James Renwick
- >
- > [https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low\\_cost\\_deterministic\\_C\\_exceptions\\_for\\_embedded\\_systems.pdf](https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf)
- >
- > Freestanding?
- >
- > 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: 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
- >
- >
- > \*Apr 13, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- > \*May 11, 2022 02:00 PM ET/1800 UTC: Games
- > \*June 8, 2022 02:00 PM ET/1800 UTC: Embedded
- > \*Jul 13, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- > \*Aug 10, 2022 02:00 PM ET/1800 UTC: Games
- > \*Sep 12, 2022 02:00 PM ET/1800 UTC: Embedded
- > \*Oct 12, 2022 02:00 PM ET/1800 UTC: Finance/Low Latency
- >



## Minutes for 2022/06/08 SG14 Conference Call

Meeting notes.

Thanks to Rene Rivera for chairing

On Tue, Jun 7, 2022 at 9:11 PM Michael Wong <fraggamuffin\_at\_[hidden]> wrote:

- > Topic: SG14 Low Latency Monthly This meeting is focused on Games
- >
- >
- > Hi,
- >
- > Michael Wong is inviting you to a scheduled Zoom meeting.
- >
- > Topic: SG14 monthly
- > Time: 2nd Wednesdays 02:00 PM Eastern Time (US and Canada)
- > Every month on the Second Wed,
- >
- > Join from PC, Mac, Linux, iOS or Android:
- > <https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJuT1loeXpKbTcydz0>
- 9
- > Password: 789626
- >
- > Or iPhone one-tap :
- > US: +12532158782,,93151864365# or +13017158592,,93151864365#
- > Or Telephone:
- > Dial(for higher quality, dial a number based on your current location):
- > US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1
- > 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833
- > or 877 853 5247 (Toll Free)
- > Meeting ID: 931 5186 4365
- > Password: 789626
- > International numbers available: <https://iso.zoom.us/u/abRrVivZoD>
- >
- > Or Skype for Business (Lync):
- > <https://iso.zoom.us/skype/93151864365>
- >

> 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

>

Rene Rivera

Andrew Drajeford

Arthur O'Dwyer

Kim Nilsson

Staffan Tjernstrom

Arthur O'Dwyer

Matthew Butler

Timur Doumler

Michael Wong

Henry Miller

Chaeley Bay  
Ronen Friedman  
Jens Maurer  
Tristan sizemore  
Ka ming Chan  
John MacFarlane

- >
- > 1.2 Adopt agenda
- >
- > 1.3 Approve minutes from previous meeting, and approve publishing
- > previously approved minutes to ISOCPP.org
- >
- > 1.4 Action items from previous meetings
- >
- > 2. Main issues (125 min)
- >
- > 2.1 General logistics
- >
- > Future meeting plans
- >
- > \*June 8, 2022 02:00 PM ET: Games
- > \*Jul 13, 2022 02:00 PM ET Embedded
- > \*Aug 10, 2022 02:00 PM ET: Finance/Low Latency
- > \*Sep 12, 2022 02:00 PM ET: Games
- > \*Oct 12, 2022 02:00 PM ET: Embedded
- >
- > 2.2 Paper reviews
- > Discussion on Embedded:
- > Review latest mailings:
- > P2532 Removing exception\_ptr from the receivers concept
- > Based on the last meeting and the discussions here.
- > P2544 C++ Exceptions are becoming more and more problematic
- > We might want to chime in here.
- > /Paul
- > P. S. P2327 de-deprecating volatile received a "consensus" straw poll.



- >
- >
- > Discussion on Low Latency/Finance topics
- >
- > <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p1839r4.pdf>
- >
- > Patrice's paper on games.
- >
- > P2300
- > Swift
- >
- >
- >
- > Discussion about Games topics:
- >
- > P2388R1 - Minimum Contract Support: either Ignore or Check\_and\_abort
- > <<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>>
- >
- > Patrice's WIP on Games issues.
- >
- > Finance topics from July 14, 2021.
- >
- > <https://lists.isocpp.org/sg14/2021/06/0636.php>
- >
- > <https://lists.isocpp.org/sg14/2021/07/0642.php>
- >

std::hive suggestions by Arthur O'dwyer  
did a template information  
found wierdness  
D2596R0: Improve std::hive::reshape (quuxplusone.github.io)  
<<https://quuxplusone.github.io/draft/d2596-improve-hive-reshape.html>>  
rope ,but not like a deque  
can control blocks  
but why larger than Y, why is a maximum blocksize is useful? Minimum yes I  
see  
is it the memory allocator? but that is in bytes

3.2 max blocksize can cause big O times issues quadratic or linear?  
fixed sized rope

AD: max because flatmaps, restricting size of the element they could have to use a simd instruction - but why would restrict it dynamically by teh user, probably ok by an implementation

has a hidden state, vector has capacity, copy does not have same capacity.  
But what about hive

transient property of a hive

Matthew B please contact Arthur OD regarding these comments

- >
- > 2.2.1 any other proposal for reviews?
- >
- > Deterministic Exception for Embedded by James Renwick
- >
- > [https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low\\_cost\\_deterministic\\_C\\_exceptions\\_for\\_embedded\\_systems.pdf](https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf)
- >
- > Freestanding?
- >
- > 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

>

Embo happened.

>

>

> - Financial/Trading chairs: Staffan Tjernström

> Carl Cooke, Neal Horlock,

>

P2590 implicit object creation

>

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

>

>

>

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

>

Intel Dounia Khaldia working on matrix operations based on AMT

BLAS paper is in SG6

Aug timeframe for Bob and Guy paper

>

>

> 2.4 Other Papers and proposals

>

discussion about RUST

14:36:29 From Charles Bay to Everyone:

<https://hirrolot.github.io/posts/rust-is-hard-or-the-misery-of-mainstream-programming.html>

14:37:06 From Henry Miller to Everyone:

rust is getting attention and we are shipping in some new things. bad programmers can write garbage code in any language, but the safety would be

nice

14:40:10 From Tristan Sizemore to Everyone:

I have heard it lacks complexity/depth for some (performance) critical components of (trading/exchange) systems. One of the larger successful cryptocurrency exchanges uses it heavily for a large percentage of their core functionality, but are switching to C++ for the reasons mentioned. I have also heard compile times can be awful.

14:40:23 From Guy Davidson to Everyone:

<https://twitter.com/h3r2tic/status/1532829918194806784>

14:40:29 From Guy Davidson to Everyone:

<https://github.com/EmbarkStudios/kajiya/blob/main/docs/gi-overview.md>

14:41:08 From Matthew Butler to Everyone:

Wow. Very smooth.

14:42:19 From Michael Wong to Everyone:

Here is one: <https://blog.sonatype.com/this-week-in-malware-may-13th-edition>

And here is another:

<https://portswigger.net/daily-swig/rust-patches-sneaky-redos-bug>

14:43:44 From Michael Wong to Everyone:

<https://openssf.org/oss-security-mobilization-plan/>

14:45:37 From Arthur O'Dwyer to Everyone:

[https://en.wikipedia.org/wiki/Politician%27s\\_syllogism](https://en.wikipedia.org/wiki/Politician%27s_syllogism) :)

14:46:53 From Charles Bay to Everyone:

"Rust is mainstream like NodeJS is mainstream. NodeJS has less than 2% marketshare globally. Rust has 0.01% marketshare. Thus Rust is even less "mainstream" than NodeJS by a factor of ~200. For comparison, C++ still holds 23% marketshare. (Disclaimer: Percentages here are from a series of Google searches. Broad marketshare numbers are never accurate but can be useful for providing some perspective.)" ...from:

<https://developers.slashdot.org/comments.pl?sid=21483756&cid=62601430>

14:51:37 From Kim Nilsson to Everyone:

In my community the people are just waiting for the green light from above to start using Rust. Generally speaking I am seeing a strong correlation between fans of (hyper) modern C++ and fans of Rust.

14:52:26 From Matthew Butler to Everyone:

The use of UB and ill-formed, NDR is a real killer for us on safety.

14:53:18 From John McFarlane to Everyone:

A lot of confusion around this: \*nobody\* advocates use of UB.

14:54:48 From John McFarlane to Everyone:

But for a formal model of what counts as a bug, either UB or something v. similar is invaluable.

14:58:44 From Kim Nilsson to Everyone:

Have to leave. Bye everyone!

15:00:26 From René Ferdinand Rivera Morell to Everyone:

<https://ziglang.org/>

15:00:41 From Timur Doumler to Everyone:

I know some embedded people who are huge fans of Zig!

15:00:47 From Guy Davidson to Everyone:

<https://ziglang.org/>

15:01:18 From John McFarlane to Everyone:

<https://ziglang.org/learn/overview/#performance-and-safety-choose-two>

15:01:27 From Timur Doumler to Everyone:

Michael, you should ping David Sankel and ask him to give you his slides or a recording of his talk or something

15:01:38 From Timur Doumler to Everyone:

"Rust features I want to have in C++"

15:04:40 From Henry Miller to Everyone:

if you can't find someone I'll ask my boss about going, right now all travel still has to be approved but that should change

>

>

> 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
  - >
  - >
  - > \*Apr 13, 2022 02:00 PM ET Finance/Low Latency
  - > \*May 11, 2022 02:00 PM ET: Games CANCELLED
  - > \*June 8, 2022 02:00 PM ET: Games
  - > \*Jul 13, 2022 02:00 PM ET Embedded
  - > \*Aug 10, 2022 02:00 PM ET: Finance/Low Latency
  - > \*Sep 12, 2022 02:00 PM ET: Games
  - > \*Oct 12, 2022 02:00 PM ET: Embedded
  - >

## Minutes for 2022/08/10 SG14 Conference Call

On Tue, Aug 9, 2022 at 10:25 PM Michael Wong <fraggamuffin\_at\_[hidden]> wrote:

> Topic: SG14 Low Latency Monthly This meeting is focused on Embedded

>

> Hi,

>

> Michael Wong is inviting you to a scheduled Zoom meeting.

>

> Topic: SG14 monthly

> Time: 2nd Wednesdays 02:00 PM Eastern Time 1900 UTC (US and Canada)

> Every month on the Second Wed,

>

> Join from PC, Mac, Linux, iOS or Android:

> <https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJuT1loeXpKbTcydz09>

> Password: 789626

>

> Or iPhone one-tap :

> US: +12532158782,,93151864365# or +13017158592,,93151864365#

> Or Telephone:

> Dial(for higher quality, dial a number based on your current location):

> US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1

> 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833

> or 877 853 5247 (Toll Free)

> Meeting ID: 931 5186 4365

> Password: 789626

> International numbers available: <https://iso.zoom.us/u/abRrVivZoD>

>

> Or Skype for Business (Lync):

> <https://iso.zoom.us/skype/93151864365>

>

> 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
- >

> Timur, Andrew L, Charles B, Detlef V, Jake Fevold, John McF  
>  
Ka Ming Chan  
Mattew B, Patrice, Staffan, Michael, ROnen Friedman

- >
- > 1.2 Adopt agenda
- >
- > 1.3 Approve minutes from previous meeting, and approve publishing  
> previously approved minutes to ISOCPP.org
- >
- > 1.4 Action items from previous meetings
- >
- > 2. Main issues (125 min)
- >
- > 2.1 General logistics
- >
- cppcon meeting
- feedbacks, audio delays
- experiment at cppcon may be different than what is at std meeting
- laptop mikes,
- headsets
- Patrice as co-chair
- Just an afternoon session



can attend fully remote, and not cppcon reserved  
do we still have cppcon physical reservations (paid)

goal:

1. allow games people to submit paper at cppcon
2. allow cppcon people to experience a std meeting
3. simulate a hybrid meeting for future purposes

>

> Future meeting plans

>

> \*June 8, 2022 02:00 PM ET: Games

> \*Jul 13, 2022 02:00 PM ET Cancelled

> \*Aug 10, 2022 02:00 PM ET: Embedded

> \*Sep 14, 2022 all day: CPPCON

> \*Oct 12, 2022 02:00 PM ET: Finance/low Latency

>

>

> 2.2 Paper reviews

> Discussion on Embedded:

>

Accessing object representations by Timur, looking for wording experts

<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p1839r5.pdf>

>

> Review latest mailings:

> P2532 Removing exception\_ptr from the receivers concept

> Based on the last meeting and the discussions here.

> P2544 C++ Exceptions are becoming more and more problematic

> We might want to chime in here.

> /Paul

> P. S. P2327 de-deprecating volatile received a "consensus" straw poll.

>

>

> Discussion on Low Latency/Finance topics

>

> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p1839r4.pdf>

>

> Patrice's paper on games.

>

- > P2300
- > Swift
- >
- >
- >
- > Discussion about Games topics:
- >
- > P2388R1 - Minimum Contract Support: either Ignore or Check\_and\_abort
- > <<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>>
- >
- > Patrice's WIP on Games issues.
- >
- > Finance topics from July 14, 2021.
- >
- > <https://lists.isocpp.org/sg14/2021/06/0636.php>
- >
- > <https://lists.isocpp.org/sg14/2021/07/0642.php>
- >
- > 2.2.1 any other proposal for reviews?
- >
- > Deterministic Exception for Embedded by James Renwick
- >
- > [https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low\\_cost\\_deterministic\\_C\\_exceptions\\_for\\_embedded\\_systems.pdf](https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf)
- >
- > Freestanding?
- >
- > 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: 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)
- >
- MW to put out a call for paper for cppcon sg14
  
- >
- >> 5. Closing process
- >
- > 5.1 Establish next agenda
- >
- > 5.2 Future meeting
- >
- >
- > \*June 8, 2022 02:00 PM ET: Games
- > \*Jul 13, 2022 02:00 PM ET Cancelled
- > \*Aug 10, 2022 02:00 PM ET: Embedded
- > \*Sep 14, 2022 all afternoon 2-5 pm ? Denver Mtn town: CPPCON
- > \*Oct 12, 2022 02:00 PM ET: Finance/low Latency
- >

## Minutes for 2022/10/12 SG14 Conference Call

Thanks to Timur for chairing this Low Latency session.

Michael Wong

Guilherme

Ronen Friedman

Guy Davidson

Timur Doumler

John McFarlane

Tristan Sizemore

Ka Ming Chan

Detlef Vollmann# CH

Piotr Grygorczuk

Vishal Oza

Sam Obeng

René Ferdinand Rivera Morell {US - C++ Alliance} (René Ferdinand Rivera Morell)

Brett Searles

Henry Miller

Jake Fevold

14:01:07 Okay, everybody, it's. i'm actually waiting for let me see, i'm actually waiting for teamer to call in because he that he would share this.

14:01:24 Oh, yeah, there he is. excellent. Timu is in

14:01:32 Hello, everyone! Can you hear me? Yes, we can. Yes, Hi!

14:01:38 Guy. Hi, Teama, Thank you for calling in i've made you Co-host.

14:01:42 Okay, great, you should be able to control everything so yeah yeah just I'm, not.

14:01:49 I haven't done this before on this particular study group so I'm not sure. I should probably keep an eye on the conversation.

14:01:55 Make sure stuff doesn't overrun make sure people don't interrupt each other.

14:01:59 That kind of stuff Yeah, that's about it and I've sent an agenda, but it's mostly loosely based on the idea of up of what we've done before.

14:02:09 So. I think this is mostly a discussion about the low latency aspects. Given that this is this month's topic.

14:02:20 And We were also going to do the games called would put trees, but he can't call.

14:02:25 He cannot call in today, so yeah so we'll be mostly low latency, and you can.

14:02:31 You can close it whenever you think you're done yeah yeah I I am recording the call, and I'm also recording the transcript.

14:02:40 You can if if if somebody on you know can still take some loose notes that would be fine as well, not, then I'll just take it from the transcript, and that way Yeah, Do you think we should Still, have a

14:02:53 proper scribe. for the session it'd be good if somebody could do a little bit of scribing.

14:03:00 But like I said I've kind of automated enough that I can kind of take on the transcript.

14:03:04 It's the transcript won't be great because the machine translation will look really strange sometimes, because as you can see right, The The machine translation literally takes everything.

14:03:15 Okay. So then, yeah. I was under the impression that you would be discussing Patricia's paper.

14:03:21 So now that this is not happening yeah he's not here.

14:03:26 So I I send an agenda out last night you can just take that agenda.

14:03:28 I'll take yeah, you can share screen on that. Did you want

14:03:35 I actually have to disappear briefly to go to

14:03:38 My fourth covid like my fourth covid shot I forgot that I've scheduled it doing this call, but it just means that I can still call

in. I will still be listening in and just in case something goes

14:03:49 wrong. But you should be okay. Let me see I'm just gonna make sure I might have put given the the sharing to the room.

14:03:57 Okay, Okay, So you should be able to see the agenda.

14:04:04 Now, yeah, I want to. a team of the co-host.

14:04:05 Sorry, Samuel, bang I I gave it. I took. Yeah, I was about to remind you.

14:04:12 No, no, nothing, nothing not nothing personal. I'm sure you'll find as a goal, too.

14:04:17 But I just wanna make sure that a few people have co-hosts just in case something goes wrong.

14:04:21 I mean

14:04:30 Can you all see my screen, or rather the agenda?

14:04:35 Yes, Okay. great. So Then I don't know Michael, before you go away.

14:04:41 One last question: Is there any formal ceremony around like the code of conduct?

14:04:47 And all of that, because look at it at their leisure.

14:04:54 The only thing is we probably do have. We should record down the names of people who actually are here, and then

14:05:02 The Yeah. but but but i'm not to worry about that, because zoom is gonna record all of that, anyway.

14:05:10 And The only thing I would say is that there's a little bit of a logistics I guess when you get you know

14:05:18 So what we we can go to Section 2.1 right away.

14:05:22 Then I can just drop off and then i'll i'll come back on after I get to my destination.

14:05:27 Okay, 2.1. The logistics. So Cpp: Con did happen. A number, you guys were there so you can give your impression of it.

14:05:36 The minutes minute, and but because it's a face to face it actually goes on the wiki whereas these minutes, because the virtual I actually send them out to the reflector I know it's a bit of a

14:05:47 weird disconnect but that's that's been the convention in the past, where face to face goes on the wiki face to face. minutes goes on.

14:05:57 The wiki, and then the the virtual minutes gets sent out

14:06:01 So literally, So basically the whole world can see it in that way.

14:06:04 But that's really about it I think We're still on track for our face to face meeting in Cona.

14:06:13 I don't anticipate having an idea\* meeting there unless someone wants to.

14:06:18 I will probably be leaving on Friday to go to my next meeting.

14:06:22 But you know there's there's you know you guys can, can can talk about that logistics a little more.

14:06:30 Yeah, It would be interesting to hear who here is anticipating being an kon in person.

14:06:37 So I I will be there I was see Michael you said You're gonna be there Okay, I don't see any other hands.

14:06:47 But yeah, I think that's interesting. Oh, jake are there any people who will be participating remotely

14:07:03 I I might drop into C 6. if if there's an meeting. I have There is a meeting because we're going to be discussing linear algebra.

14:07:12 There's new a new version of the paper is Well, i'm going to make sure it's ready by by by mailing deadline.

14:07:20 You mean staying up all night, but i'm very nearly ready on any new version, so I hope Sc.

14:07:24 6 will come, being otherwise, i'll be quite crumpy

14:07:30 Alright, should we should we find the scribe? Is anybody willing

to take notes?

14:07:34 Or do you all think that the Zoom automated thing is enough

14:07:43 I actually think the zoom is okay? i'm not sure how many people actually read minutes.

14:07:48 You got to Frank normally. you know it's it's discussing points on papers.

14:07:53 Yeah, I do read minutes from something like you know core telecoms or the the plenary, or you know Bsi meetings as well, but probably not these, because like there's a lot of kind of casual conversation going on I

14:08:05 guess we're just going to be captured in zoom anyway.

14:08:11 Okay, so you can also see the agenda role call of participants.

14:08:16 I don't know. If do we do this here, yeah you can go ahead and do that if you want alright.

14:08:24 So like everybody introduces themselves briefly as that it Okay, all right.

14:08:28 I'm gonna go with the order that I have here on my screen.

14:08:37 First one is guy

14:08:38 Yeah, hello, everyone, My name's God davidson's head of engineering practice.

14:08:41 You created assembly. i've worked for 23 years on the Total world franchise.

14:08:43 Everyone should go and buy a copy of every game right now.

14:08:48 I work on the I I 7 c\* committee I'm.

14:08:52 At the moment trying to stuff linear algebra into the standard as soon as possible, adding this to C.

14:08:57 \* cut off alright, Michael right my name's Michael wong i'm the

14:09:05 I guess the chair of a number study groups as well as part of the Directions Group.

14:09:09 I've been in C. for about 25 years now.

14:09:13 Almost a quarter century. So thank you for joining

14:09:16 Thanks, Michael Giller. Hi! can you hear me yeah I'm.

14:09:24 A research engineer at Ait Austin Institute Technology.

14:09:27 And this is my first time here so yeah so sorry for the notes zoom doesn't show us your last name.

14:09:34 What's your what's your full name for the notes the Glenn car.

14:09:38 Those are the Soza hutrigues it's a very long name.

14:09:40 So how could you? Maybe Ronan

14:09:50 Yes, yes. do you hear me? Yes, hey? i'm writing software for a red hat short, and in a short while for will be writing for lbm currently in the last few years, involved in a large project the storage area before that many using

14:10:16 embedded the umishal yeah so I would consider myself a C. Geek kind of have been going into the standards, and you know, listening to these committee meetings.

14:10:37 Haven't gone there for come here for a while but i'm interested in

low latency finance embedded systems and games.

14:10:45 Thank you. Sam. Hi! My name is Samuel Bay, and I just finished my Mc.

14:10:55 Asian intelligence and invasion before then. I did about several years of software before going to school. Currently, i'm done with school, and I am yet to get my new job.

14:11:05 But I love C, and I have worked with it, and I love B here so.

14:11:11 Thank you. thank you Renee yeah let's see i'm one of the sub-chairs or games in this group.

14:11:23 And also frequent the bunch of the other groups i'm into tooling.

14:11:28 I work for a game development company called disbelief I'm.

14:11:35 Also board member of the C alliance i'm a contributor to boost.

14:11:42 I have my own various libraries in Github. yeah.

14:11:48 I think that's all. Thank you Jack Yeah. my name is I work for Boomer to lp.

14:12:01 Thanks. John John Mcfonin. hi there I know a joke Phone: i'm a member of I think on the Thank you.

14:12:21 Tristan, hey? I just joined this as she put sorry we can't hear you very well.

14:12:37 Could you possibly turn up your volume a little bit?

14:12:40 Can you hear me now? yes, so hi guys i'm kristin I just joined this Sg\* group.

14:12:55 Via email request I asked Michael Wong I could join.

14:13:02 I've just joined this observer I c.

14:13:15 Professional work at a company called Butt do crypto trading.

14:13:28 It's a mobile app i'm just observed

14:13:37 Thank you. We have 2 other people who joined the call.

14:13:41 I ha probably not going to pronounce this properly. apologies.

14:13:46 Coming, Chan: yeah, hi there, you put, Nancy is very correctly, actually.

14:13:54 Okay, do you? wanna tell us a little bit about yourself.

14:14:00 Oh, yes, sure. I i'm actually sitting in London I worked in the finance area.

14:14:08 Hi Co. he would get for a hedge fund.

14:14:11 Call Pauli asked me, which is a actually a us hedgehog.

14:14:17 So I use c in quant quantitative library.

14:14:25 I in general interested in the low latency and actually everything else as well.

14:14:33 So hopefully I would be able to contribute something. But then

14:14:37 At the moment I could I just want to listen in and the I'm.

14:14:42 Sure it is about what you guys are developing yeah that's just myself.

14:14:48 Thank you, thank you, and and finally Deadlif. Oh, my name is Tetra Foyman, i'm from Switzerland, however, mainly in embedded. But right now I also have a go check that has Low latency



14:15:06 in the sense that I have to do I have to boot. something from an embedded system to a server. and that has to be done at the 6 deadline that's for me.

14:15:22 Thank you, and we have one more person who just joined.

14:15:26 Pyoto. Yeah. waiting for them to connect to audio

14:15:37 Pyotr! Can you hear us? Can you hear me

14:15:46 Hello, Filter. Hello, So sorry just to connect it now.

14:15:55 And my microphone didn't work so

14:16:05 So my name is Peter and I was part of those calls before just to join today.

14:16:14 And I work for intel in the embedded systems.

14:16:19 So yeah, thank you. My name is Steve Modumna.

14:16:25 I am developer advocate at japanese so we make tools.

14:16:28 So i'm into tooling I I also spent a lot of time working in audio particular music production software, which is another low agency, application, because it's kind of real-time number crunching.

14:16:41 So I'm very interested in low latency and real-time applications of C.

14:16:46 And i've also been active on the c committee since I think 2,016.

14:16:51 So yeah, I there's seems to be a couple of people here on the call, who have not been here before so welcome.

14:16:57 And thank you very much for joining i'm just gonna go through the agenda here.

14:17:02 Which I got from Michael. action items from previous meetings I don't know what that is.

14:17:10 If you Michael is still on the call we don't have any you don't have any.

14:17:16 You don't have any okay great so you already discussed general logistics.

14:17:19 We had a meeting at Tp. call and the minutes are here on the wiki

14:17:24 I I'm literally just reading out the minutes here sorry the agenda here.

14:17:28 So we do have future meetings planned.

14:17:32 You will not have a call in november because we're gonna be meeting face-to-face and Cono.

14:17:38 At least some of us will. however, there are calls planned for December, January and march at the dates in the agenda, and they're going to be continuing rotating games embedded and

14:17:51 finance national latency. Michael, do you have anything to add on that?

14:17:56 Yeah, we could probably at this point stop saying that we have we're gonna be doing more stricter.

14:18:01 Membership attendance and because of that I think everybody has to choose meeting number, which is, unfortunately, but there are ways that we can get around it, especially alliance. So maybe you can talk a little bit

about that rules change do you know what

14:18:20 i'm talking about teaching Yes. So so from my understanding

Basically, if you want to participate in a committee meeting, including a telecom, including study group telecons like this one.

14:18:35 You basically have to be on the Iso Global Directory, or you have to be appointed as a delegate or as an alternate by your national body.

14:18:45 So if you are not on that iso delegates list, then you can attend one meeting as a guest, and after that you should get into touch with your national body.

14:18:57 And become a member of Iso by a delegate of Iso

14:19:03 So. I guess that means this already applies to this telecon.

14:19:09 Am I right? No, not exactly we're gonna have some we're gonna have a bit of a grace student in particular is kind of different.

14:19:18 It's always been an open outreach group at which people did not have to be official members. but now that is changing. and so I have to find a way to as well for this i'm i'm i'm

14:19:32 i'm basically just putting off but I think that there's a way it's not easy for obviously for people to join to become members.

14:19:43 So there's a way at which you can join and become A member would be plus alliance. Now send some. Maybe he can tell us a little bit about that I don't know if they can do that for us.

14:19:58 It's actually the c foundation not the supposed to no.

14:20:08 So. so actually, a guy has his end up. Yeah. Sorry.

14:20:16 Excuse me. Yeah. there are 2 organizations which are offering membership to people such that they become members of the Iso.

14:20:25 They they can record in the iso directory there's the cpus foundation, also boost foundation.

14:20:30 You can write to other organizations. request to be to join, and you will be covered.

14:20:39 As members of those organizations through insights.

14:20:43 Renee the I don't know if the Boost Foundation has actually finalized their procedures for that.

14:20:52 I know they have. they have a request for it and the board met, and they're okay with it.

14:20:58 But they I don't think they've actually gone through and actually become a member.

14:21:02 Okay, I will. I will say with David sample leave.

14:21:06 It was all good. But yeah let's let's wait until this action in that in the public domain there has been an announcement in the public domain about the sequence Foundation. there.

14:21:16 So you can contact Actually, you just write to have such a sutter at gmail dot com shots or also nina nina.

14:21:29 Rans, Who is the the secretary, I think of the or one of another member of the Yeah.

14:21:38 Either, either. Herborneina, contacting them will do.

14:21:42 Yes, it's it's the 9 situation.

14:21:50 But this is the thing that happens with international bureaucracy.

14:22:01 Let me just

14:22:08 Yeah. So if any of you want to keep participating.

14:22:12 And these calls and want to become a member. Please contact one of those organizations.

14:22:21 All right. I think we can now then move on to to paper reviews, unless there's any other kind of bookkeeping or general logistics business.

14:22:37 Can someone sum up if in a few minutes the face to face meeting for those of us who cannot read the this link?

14:22:48 Sure. Maybe I can like do a little of pubits.

14:22:52 And then other people who were there can like fill in the blanks.

14:22:56 So we had I think 3 papers one was

14:23:02 I had a paper on kind of reading the bits of an object. it's like reading object representations paper P.

14:23:10 1839, I think, which we discussed because it's relevant to the latency.

14:23:17 That was the very long discussion. Then we had quite a long discussion about at least. voice paper, which is basically not not the published paper yet, but more like a big word document with lots of ideas future proposals that are relevant

14:23:37 for low latency and these kinds of applications. So there was a lot of discussion around, like all the kinds of features that were in that document that some people thought might be a good \*.

14:23:51 And We had a third paper which I can't recall now, maybe somebody else remembers what that was

14:24:00 Or I can just cheat and look into the minutes and then probably i'll remember

14:24:11 Oh, yes, there was a paper about graph data structures which I wasn't in the room for so I can't really comment on that.

14:24:21 Oh, I think I was in a room but I wasn't really following this question on that one very well

14:24:31 Yeah. Does anybody else have any more info about that?

14:24:38 Or does that answer like roughly what was going on there?

14:24:41 I think the the other thing that was going on there that was really interesting is that that was the first of meeting of a committee study group where we try to do hybrid attendance where we had a bunch of people in the

14:24:53 room, and then also a bunch of people joining online at the same time, And which is something we want to do in Coney.

14:25:00 And this was kind of the grand rehearsal for that, and we had quite a few problems with everybody hearing, like everybody on line, hearing what the people in the room were saying, and things like that.

14:25:11 So I didn't go very smoothly but I think it mostly worked. and it

gave us kind of good kind of good input on on what to improve for kona, because for corner there's gonna be a lot of sessions running in this  
14:25:24 kind of hybrid mode where you can attend remotely, or you can be in the room.

14:25:29 So that was a bit of an experiment which I think was successful.

14:25:32 Maybe not in a sense that it went very smoothly, but in a sense that we we got valuable data on on how to pull off such a hybrid meeting.

14:25:39 So hopefully in Kona. it's going to be smoother than it was there.

14:25:51 Alright. So then, on the agenda, we have a bunch of papers.

14:26:01 We have P. 2, 5, 3, 2 which is removing exception.

14:26:07 Pointer from the receiver's concept does anybody wanted to present something about that, or

14:26:22 I don't I don't think there's authors for any of these papers present.

14:26:26 I think today there's just going to be a discussion about low Lane Sea.

14:26:29 That's all that's been being discussed on the reflectance in the last couple of weeks.

14:26:35 I think you had some you had. you have started the discussion thread and maybe I think what we need to do is just summarize that for everybody on what's been What's been discussed so far in case Well, for people who

14:26:49 don't follow the the reflected easily I don't know if that helps.

14:26:53 Okay, yeah, i'm not sure what discussion this this relates to in particular.

14:26:57 So I'm not sure if anybody else here knows what this is about and could maybe talk about that

14:27:10 Yes, but 2 messages on the on the reflector in the last 2 weeks, I believe

14:27:19 Okay, could you potentially. that's something that we can screen share and look at or

14:27:34 Which kind of not sure how to to conduct this discussion because i'm just not aware of this particular i'm I don't know what what emails you're referring to

14:27:49 Actually, I I believe. Michael is referring to your original email with the request for financial, about their low latency, or how real time requirements.

14:28:08 Oh, that Okay, yeah, yeah, yeah, Yeah. Okay, that's obviously i'm aware of because I I wrote this email wasn't related to any kind of proposal.

14:28:18 So I I sorry I just didn't realize that this is what this is what you were talking about.

14:28:23 But sure I can talk about that. Yeah, I I can definitely talk about that.

14:28:28 And then the other email what's the other email i'm just looking

at the reflector right now.

14:28:42 Just trying to figure out like what's the agenda basically for today.

14:28:45 So we can talk about the the email thread that I started

14:28:50 And then, basically, is there anything else that that you should?

We should talk about

14:29:00 I have a feeling it might have been stefan's response to email as well.

14:29:06 The whole thread was quite interesting. Okay, so i'm just so, then.

14:29:12 So so that's then just talk about that thread and if there's no other topics.

14:29:16 Then we can join after that. So yeah, I can. let me.

14:29:20 Then I can actually share screen share and and bring that email up here.

14:29:25 If if you like. just give me a second please sorry i'm a bit.

14:29:33 I'm a bit slow today. it's been it's been a long day, and it's late in the evening.

14:29:38 Here. Here we go! Hey! Let me share this.

14:29:57 Can you? Can you see my screen with an email thread on it?

14:30:02 Yes, okay, great. So just a little bit of context, what this was about.

14:30:07 So I was actually due to give a talk. Actually, that guy Davidson has invited me to at his company about low latency, c.

14:30:16 And as I was preparing this talk, i'll sorry debt that you have your hand up, No, okay.

14:30:25 So, as I was preparing this talk it, was kind of about kind of typical properties of low latency low latency applications, and how I wanted to figure out how in which ways they're similar and which way is that

14:30:38 different, so obviously in low latency whether it's games or finance or audio processing or awesome.

14:30:48 Also quite a lot of embedded use cases. Not only do we care about whether a piece of code does.

14:30:54 The thing it's supposed to do but also how fast it does it right?

14:30:57 So like. There is kind of an implicit kind of deadline, or we want to get the answer faster.

14:31:01 We care about latency. So so I was kind of looking at

14:31:06 Where different use cases differ. So, for example, in finance, versus an audio was in gaming.

14:31:15 You have kind of different different time scales there, right? so in.

14:31:18 So I was kind of familiar with audio very much, and not so much familiar with

14:31:23 Maybe a little bit familiar with games, but not so much familiar with high frequency trading in particular.

14:31:28 Yeah, and I I just so sorry. I think like those 2 portions of the

high frequency training.

14:31:36 I think there's one that's low latency and then there's one that's high throughput where what what you're doing is you're kind of the hype group but you don't necessarily care

14:31:45 about time. You kind of want to push all of the data.

14:31:48 The high frequency training would be more towards towards like a fpgas or solar flare cards where you what you're dealing with is you want to when you're getting information from the exchange you want to

14:32:01 react to that as fast as possible, whether that's order information or whether that's market data, you want to be able to to at least put some some state on your State machines.

14:32:16 So that, and maybe have some code that might execute

14:32:22 That might be more towards the high frequency. trading as far as I kind of had experience with.

14:32:29 Yes, yeah, I think that's accurate like I think if you're in finance. i'm not a finance person, but I guess you have lots of applications where you just kind of crunch.

14:32:37 Lots of numbers you you I don't know run some risk models or whatever you do there. And then it's just a lot of kind of processing we care about throughput, because there's a lot of data coming in Yeah, that that that

14:32:48 can be one thing there's also the other side side where you you basically want to react.

14:32:52 So, for example, let's say that you find like a you know a company price changes, or like you know, a future or commodities price change.

14:33:03 You want to be able to react as soon as you get that portion that's kind of a different portion.

14:33:08 That's more the low latency, stuff where high frequency is basically you you're getting all of the number crunching in your you're forwarding that into you know a a you know container that can you know Do all

14:33:21 the processing. Yeah. So So I was specifically looking at use cases where we care about low latency and not high throughput.

14:33:29 So it seems like that kind of orthogonal aspects of performance in a way right?

14:33:33 No no that that's why you kind of shouldn't say high frequency training.

14:33:38 It's low latency. low latency processing rather than a high frequency training.

14:33:44 Usually that's and that's, usually related to more or less algorithmic trading due to the fact that you want your algorithms to respond as fast as possible to, or at least in a controlled way depending on if you're dealing

14:33:56 with one exchange, or if you're dealing on with many exchanges.

14:34:00 You want to do that? you know, you know and they small if you're

dealing with like one very fast exchange, where you have all your your servers, or your you know your order routers, etc.

14:34:17 You know, very close like, you know, no more than a couple of feet away from the Exchange.

14:34:21 You know that's going to also help you because you don't have to worry about the lag Now, if you're kind of placing those orders on a bunch of different exchanges.

14:34:32 You might have to kind of you know. you know, work with the lag to to make sure that you, you kind of have your orders delivered at multiple exchanges.

14:34:42 So like you worst case would be that you didn't have the you know the the slower exchange react about at the same time as the higher exchange that's like 1 one stuff that that's like one way you can do

14:34:56 that. if you're just doing something like arbitraging right yeah, that's interesting.

14:35:05 So a bunch of people reply to to this this thread here.

14:35:10 So Stephan wrote reply: Wesley, author And and from what I kind of learned from that, is that kind of low latency.

14:35:23 Financial applications seem to have at least 2 interesting properties.

14:35:26 That kind of distinguish them from all other low latency applications.

14:35:29 I am aware of, like audio processing or gaming.

14:35:34 One is the time scales, whereas, you know, games or audio operates usually kind of in the order of maybe one or 10 kind of milliseconds.

14:35:45 Finance seems to be operating on much, much shorter time scale.

14:35:48 So are you like, and and kind of certain high, free, concentrating scenarios.

14:35:52 You want to. if you get information from the exchange, you want to then send something there within an order of a microsecond or even even faster.

14:36:02 So I thought that was quite remarkable. And the other thing that I found interesting is kind of the deadline you have, where, for example, in games and in audio processing, you know what you deadline is right?

14:36:13 So you have a certain frame rate for example if you're producing video, or you have some kind of audio buffer size.

14:36:18 If you're producing audio and you kind of know that you have to produce data within for example, 1 ms or 10 ms in order to not calculate.

So in order not to drop a frame so you can

14:36:32 know what latency is expected of you in order to you know get the desired result, Whereas, these high frequency trailing up applications have this curious probability that you don't actually know exactly what your deadline is you obviously need

14:36:44 to be faster than all your competitors. But you don't know exactly how fast that is.

14:36:50 So you in the end, like kind of just try to be as fast as possible, which is also kind of quite unique.

14:36:57 In this kind of wider field from from kind of my perspective which I found very interesting, Right?

14:37:18 Basically it's you know you do although I think like sometimes there's also the idea of being kind of slow If you're dealing with multiple exchanges simultaneously and placing those orders basically the idea is to get you

14:37:22 know. Get your orders you don't know exactly what time and I think also, another thing is networking high frequency trading usually has some sort of networking issue.

14:37:28 That's kind of why? you don't know the time scale. Because you basically have to worry about how the exchange goes and processes those orders.

14:37:37 If those the exchange is order management system is very slow, then you know you can't really do anything about that other than wait.

14:37:47 And you don't know about that yeah they can they can always change, you know, change their their order book or their order processing structure over over time either to slow it down to deal with like you know, certain hardware issues

14:38:02 or, you know, make it reliable, or maybe for some sort of regulation purposes, or something like that.

14:38:14 Right? Yeah, that's interesting

14:38:24 Yeah, So basically, basically, the whole point I think you're talking about is not necessarily high frequency trading.

14:38:31 But low latency or algorithmic trading, which really is about like reacting to the orders.

14:38:39 And I think, you know, I don't think anyone does like audio processing or embedded systems.

14:38:46 Truly too much on the network. Usually they're on the local device and kind of in an enclosed area where I think high frequency trading or algorithmic trading is more on the more on the network.

14:39:00 It. It may be something that may be processed on the cloud, or It may be something that's processed, you know, in in some sort of embedded system, or sorry not a or like a network.

14:39:16 A you know just a plain old Internet connection or you can you have stuff that's related to web software.

14:39:23 It's that might be not necessarily the same as like a true socket connection.

14:39:30 Then you also have it connections that also also work on that type of system.

14:39:39 So. Yeah, that's interesting, I think it's not quite right that other low latency of the applications don't work with networking.

14:39:46 Obviously you have audio and gaming where something like this app zoom. right is is, you know, a real time application that uses networking,



or you know, in games here you have networking as well, it's just that I think

14:39:58 the time scales are different, because you were talking about tens of milliseconds, and I'm actually curious to hear about.

14:40:03 You know. Maybe there's some gaming people like die, for example, who could talk a little bit more about the kind of networking aspect?

14:40:10 I think that the 2 differences I can see is like the time scales, and the other one is, whether you, whether or not you control your networking stack.

14:40:18 I think please correct me. I'm wrong but my understanding is that in and trading you have your own like customized kind of network cards and customized drivers, where you kind of bypass the kernel and like some things in between you can do

14:40:33 even like with FPGAs, and and so you have like completely customized stack there.

14:40:37 Whereas if you're doing audio games you do use the network, and you do want to be fast. You don't want to have an unacceptably slow latency, you know, when you do you know I don't know playing some

14:40:48 first person shooter over the network or something but you don't it's it's just regular consumer hardware, which you don't have any control over unless you want to console maybe but still then you don't

14:40:55 know what networks. set up you have so well well I think there's 2 portions you have your own network where you're trying to minimize.

14:41:04 You're doing, you're having cards like solar flare, or you know it's custom FPGAs.

14:41:10 Some people might do like you know if you're trying to get in.

14:41:12 You might do stuff with, you know. regular conventional hardware. you know.

14:41:16 Maybe use a Gq. or something like that I mean this may be if you're a smaller shop that you know, are trying to start in into the low latency algorithmic trading or high frequency, trading.

14:41:28 They might do that. but I I just I'm saying like Yeah, you know you have 2 portions of the network.

14:41:35 You have the exchange, and then then you have actually, you have may have 3 you may have.

14:41:42 You might have a broker like a, you know, interactive broker, or something else.

14:41:45 That kind of you know, takes your orders, and sensitive to the exchange.

14:41:49 Then you have the exchange like the CME group ICE your X etc. New York Stock Exchange, where he basically you're sending the orders directly to the Exchange.

14:42:00 And then they are trying to, you know, process those orders as fast, you know, doing an order matching so structure.

14:42:09 So you basically, you have control over your side up to the

Exchange.

14:42:12 But then, you know, the exchange has to deal with when you're sending the orders to that that might be, you know, another stuff where you know the processing.

14:42:24 The issue is processing. You want to keep your side as fast as possible, but you might want you might not actually have any sort of control over.

14:42:34 Once you send the orders you could send, It could mean, maybe take, like, you know, a day, or it may take a you know, a couple seconds for those orders to be processed, or the order that you had sent to be processed

14:42:50 Where I think, like with a game, you basically, you know, once once everything you know, the system updates.

14:42:58 You know your networking system kind of is there?

14:43:02 Everyone tries to be as fast as possible, and has the same code.

14:43:06 You know. you know they they're trying to play the same game on the same, maybe slightly different hardware.

14:43:13 But, like, you know, the same game structure Everyone is a different, you know you have the buyers and sellers on on the Exchange, and that's probably one of the difference between the you know, embedded systems, and high frequency

14:43:29 trading or actually low latency trading. So sorry about that.

14:43:32 But yeah, so hi! I appreciate high frequency. can kind of be something where you're taking you're taking the market data.

14:43:40 You're processing that to maybe I think the this was like about a time.

14:43:46 I I think Stefan and I were working at the same company where we had.

14:43:52 We were using like a getting some market data, and just kind of calculating the implied.

14:43:59 This was this: wasn't something that I was doing per se. But this was something that a colleague was was doing where he was taking the the data and processing that on a nvidia gpu card to

14:44:14 calculate implied. implies and this was like a thing that was happening like once every month or so.

14:44:24 4 eyes introducing Continental exchange.

14:44:27 He was just doing that to get, like all of the all of you know, calculating the implied pricing from the from the actually price press data

14:44:48 Right, so that that was more towards the high frequency trading.

14:44:53 But you know you will latency, is us and you basically the idea is here you want in the high frequency trading year, trying to spread out a lot of orders, and you don't really care about you know if they're filled

14:45:04 or not. You just want to do you do a lot of order processing where low latency is more like a you know, you're sending out that are little bit more smart and you're trying to do it more towards

14:45:19 algorithmic structures

14:45:23 Yeah, it's interesting because I heard like from some as somebody who comes from the outside like I heard the terms high frequency trading low latency situating and algorithmic trading, or basically being used interchangeably so

14:45:37 where there's any difference between any of them right basically the idea is hybrid frequency is like you're sending a bunch of orders in a batch where low latency you're trying to I think low

14:45:49 latency is usually more closer to the out algorithmics.

14:45:53 Trading algorithmic. you're you're having more of A. You know what what you're supposed to do where low latency is.

14:46:00 Basically, you want to react as much as possible. So low latency and algorithmic trading might be kind of very much closely tied together.

14:46:11 Where high frequency is more like you know the idea of sending a bunch of batches. not necessarily knowing what you're doing, but you're trying to take advantage of of something you might have some sort of processing, but they may maybe

14:46:22 not be more of a smaller processing. Yeah, and you know a lot of stuff might be a OS kernel.

14:46:29 One thing that i'm kind of trying to do on my my side is a basically experiment with like a training system where where you're using

14:46:40 You're using a and micro kernel operating system and keeping all of your training structure.

14:46:47 In the user space very much connected, you know, once you once you initialize your your system.

14:46:57 Basically, i'm trying to look at using linux, 3, which is kind of like a micro kernel and just trying to prevent like, you know, the the kernel system from you know disrupting disrupting

14:47:11 the the software happening, although Chris is more like a experimental idea without an exchange, it's kind of worthless.

14:47:22 So pick that as you will

14:47:32 That if you're doing you know probably not in the high frequency, you might not be.

14:47:42 Yeah, you might you basically, with algorithmic trading you're trying to, you know, set like, you know, what what the conditions are for. for you know, making the order, You're not necessarily sending a bunch of orders you might send like a

14:47:56 pure pure orders. But you want to react when you have market data as best as possible.

14:48:01 At least That's from my point of view there might be other people who, who, you know, are also in the same domain, but not necessarily not necessarily there.

14:48:13 I think, one of the structures that i'm kind of trying to, you know, get myself a little bit of associated with this called Mediterranean, which uses c.

14:48:24 In order to create the algorithmic structures. But this is more like for smaller, you know, smaller smaller time of traders.

14:48:34 I I did think like there May be other people who are like you know if you're a hedge fund, or you're if you're something like Bloomberg or something you might not actually kind of use the

14:48:43 Mediterranean platform. you might actually use like you know raw data that you're getting from the Exchange, like the Cme group, ice, New York Stock Exchange, etc.

14:48:57 Nasdaq, etc. so you might be trying to do something towards towards that

14:49:11 I could have raised my hand. Okay.

14:49:18 Any more on this tedler related to this i'd be interested in.

14:49:31 Well, if you have a deadline of microseconds do we actually use parallelism to meet that deadline, or try?

14:49:47 Or do you try to do everything as fast as possible on a single core?

14:49:52 I would say it might depend it might be if you're dealing with like a high frequency trading where your number crunching.

14:50:00 Then parallelism can help if you're dealing with like you know, just sending out single orders.

14:50:05 I would say that maybe it might be better to to do it everything as fast as possible.

14:50:12 Rather than you know, doing doing stuff normally.

14:50:19 It's basically kind of it depends you have 2 different kind of strategies.

14:50:23 You have the high frequency trading strategy where you're basically sending a bunch of orders.

14:50:29 You know, a little bit mindlessly. and like hoping some of those get bills, or you know, hoping that you know you get paid, or something like that.

14:50:39 Then you have the low B Latency version work where you are trying to, you know.

14:50:45 Make sure that I mean it may be just running once on a single core, or you might have maybe a core that's dedicated to 2 different environments like you might have one that's right doing something like cheese futures or one that's dealing with oil

14:51:03 or something something else so that that's what you know that's what the idea of the hypothesis trading?

14:51:13 Yes, it's basically you have or that's what's related to the trading.

14:51:19 The the side you So yeah Thank you.

14:51:29 Interesting. Okay, do you mind if i'm remove myself sure.

14:51:35 So one thing I can add like from the audio processing perspective.

14:51:41 Is that also like if you have a like no latency, application there, where you're doing like real time processing, you don't want to be

doing any multi-shotting, there.

14:51:49 So typically you have one thread which is kind of the real time thread where you do your processing, where you have to generate like a new audio buffer.

14:51:56 Every millisecond and you don't wanna do that on multiple threads, because, you know, you have to synchronize those threads. and then you have to interact with a thread scheduler in order to do that

14:52:07 and and that's not going to have kind of a deterministic execution time.

14:52:11 And so you you can't rely anymore on kind of being below your deadline.

14:52:17 So it tends to be that you have, like in particular in this particular domain, that you have a single higher priority thread which is doing the real-time processing, and is not doing any parallel stuff at all and all the other

14:52:31 threads that deal with like the Gui, or the networking, or disk access, or whatever it is.

14:52:37 They do that kind of independently and if they need to exchange data. you'll use like a single producer single consumer lock free fifo, in order to do that. because that's wait free not just lock free So

14:52:49 you can again reason about execution. time there and this way you can kind of like, get in that data in and out of the real time fit.

14:52:56 So that's kind of how it works in this particular domain of audio processing.

14:52:59 But bunch of hands. Guy was first I believe yeah i'm I you said, that's We don't want to using threads because of the normalistic nature of switching.

14:53:13 And so on. I might be talking as a rubber shirt.

14:53:17 But do you think that asynchronous co routines my offer?

14:53:21 Determinism or sufficient determinism. if I may just directly reply to that. My understanding of core teams is that, And they don't know anything about threads, right?

14:53:35 They don't. they're not like concurrent or parallel in any way, just by themselves.

14:53:39 They're just a way of passing control to a different context.

14:53:43 Like. If you want to do parallel stuff with core teams, you have to add that on top, right.

14:53:48 By the way, you write your promise type or your whatever it is you're available, and all of that that's where the concurrent stuff or the parallel stuff goes and and that's going to use kind of the same synchronization

14:54:01 mechanisms, as is also in the language that's my understanding.

14:54:07 So you can obviously use core teams in these contexts and I think quotings are great because they're very kind of low overhead.

14:54:12 They're very efficient. but at that point you're not doing

anything in parallel.

14:54:17 If you want to do stuff in parallel you're gonna then have to add some to that synchronization mechanisms on top.

14:54:24 And then again, you're gonna run into the same problems as with all the other language mechanisms.

14:54:28 But i'm curious what other people said I think because in terms of hands we had then that left, and get out

14:54:40 Cooling is not a concurrency it doesn't know about the other fresh?

14:54:43 It just seems back and forth. Leave it with trust because there's no time

14:55:00 Up to low latency folks. What do you guys do? If you do use the Stl do you solve the determinism problem?

14:55:10 Do you just not use stology just use a version of sql that doesn't have any

14:55:19 Does anyone want to direct your reply to that yeah I can We don't throw exceptions and we minimize allocation.

14:55:29 Those the 2 greatest contributors to non deterministic behavior, c.

14:55:35 Plus plus We minute survive, not throwing exceptions that does rule that certain parts of the Stl.

14:55:44 But by by locking out allocations that if it actually eliminates all the containers, so we do have to accept some aspects of owners in which we mitigate by right well it's a creative assembly to

14:55:58 mitigate that by writing our own allocators. for example, using it, Reiner allocators using cool allocations.

14:56:05 We are allocating. all the same, you know objects the same size.

14:56:08 You can. being much more certain about you, you can place a much stronger, shorter upper band on the amount of time allocation will take.

14:56:17 Yeah, I mean audio people do the same thing. I have actually a whole talk about that called using like, I think, real-time programming with the standard library, or something like that, where like I talk about like the subset of the Stl: that you

14:56:30 can use that actually has the deterministic runtime.

14:56:33 And yeah, that eliminates everything that allocates memory.

14:56:38 It means all dynamic containers, everything that has type or Asia.

14:56:40 You can't do any of that stuff you can't do anything that kind of might have a lock inside.

14:56:47 You can't yeah So so it's it's a peculiar subset of the Stl.

14:56:52 And a lot of people they they, instead right kind of their own replacements for this.

14:56:58 So you can't use the vector because it's allocating. So you might want to write a static vector right? and actually a lot of the kind of Sd

14 proposals that we have been looking at target exactly those use case

14:57:10 and proposed facilities kind of to accomplish that, to avoid things like allocations or locks.

14:57:17 So I think debt left was was next yeah about my Oh, which no

question was already onset. But also about the Sdl: Well, about exceptions.

14:57:32 Well, if you get an exceptions, then you have something that you can't meet your deadline, anyway.

14:57:42 So that case is not really a problem, and having no exceptions, is typically pretty deterministic.

14:57:53 Time wise. So that is not a big problem. And yeah, about allocators.

14:57:59 We already heard a lot, and that's a very important thing you have to look at.

14:58:05 If you need determinism. Okay, be shy. Yeah.

14:58:14 So I was just kind of, I think, like this was about like exceptions, like I, and like, like, this exceptions and kind of like the low latency portion.

14:58:27 So basically I was just saying in the financial industry, maybe you might be able to use some exceptions or some of the exception handling, although, it, it might be kind of the case.

14:58:39 By case but basis you know. So some people, some companies might might allow them.

14:58:45 They might like. I think the the idea is like, you know you you use most of the Stl.

14:58:50 Maybe a little bit of boost and like, you know, there is some cases where you might do something you might use like the standard containers.

14:59:01 Oh, although maybe you might have to kind of you know like, you know, modify or kind of use, a different like, like, instead of using a vector you might want to use a deck or a queue in order to process, this

14:59:17 you know, push in data, and it basically because we're not necessarily looking at deterministic data.

14:59:23 We don't know when we're gonna get the information from the exchanges, or anything else. we don't.

14:59:29 I I think that in this case it might be better to have like a queue, and I think also possibly looking at stuff from the Exchange, or sorry from the from Concepts and co-.

14:59:44 Routines. That probably is also like might be useful for for people, at least on the financial low latency level compared to compared to somebody who's doing something on a embedded system, or gaming gaming thing where you can

15:00:01 write a co routine that actually handles handles like a message type that you're getting from the exchange.

15:00:11 Yeah. So can I just say one more thing about the whole exceptions thing?

15:00:18 Because somebody said, Well, if you're throwing an exception you don't care anymore about the determinism, because at that point you're like in failure.

15:00:23 Mode that's not quite true or it's not true for every use case.

15:00:28 So we know that on most platforms as long as you don't throw an exception like having exceptions in your code doesn't have runtime overhead.

15:00:37 It. It does have runtime overhead I think on windows 32 bit, but it doesn't have runtime overhead on any other desktop or mobile platform that i'm familiar with it has

15:00:47 overhead in size, binary size, so that that's important for embedded systems.

15:00:53 But more importantly, there are scenarios where the like even the arrow path needs to be deterministic like.

15:00:58 For example, if you're doing audio processing you have to call back right every millisecond you have a callback, and then you get a pointer, you have to feel like new audio frames into like the array

15:01:10 that's this pointer pointing at and that's going to be set out to the speakers.

15:01:15 So. you cannot. There, just give up on your determinism and say, oh!

15:01:20 Some exception was thrown somewhere i'm just gonna not do anything, because then you're gonna not write any data into the buffer.

15:01:25 You're gonna get an audible glitch or click which you know in the worst case, might actually destroy your speakers, because it's like a very sharp kind of like discontinuity in your waveform.

15:01:35 So if you encounter an error, you you have to do something else.

15:01:38 You have to, you know, fade out, or you know output.

15:01:44 Maybe some noise or output silence, or or do something else.

15:01:48 But you do have to deterministically produce some data.

15:01:51 Right. So you can't just give up and say Oh, an exception has been thrown.

15:01:56 I don't care anymore about this function you know like returning a result within a millisecond.

15:02:02 You just can't do that, and I imagine that There are quite a few embedded like use cases again.

15:02:08 I'm not an embedded guy but there are quite a and a imagine that quite a lot of embedded.

15:02:13 Use cases where you also have this kind of callback or deadline, where you have to get a result within x milliseconds, no matter what like I'm thinking about automotive or robotics, or you know medical kind of devices

15:02:25 and maybe there's some people here who can comment on this stuff.

15:02:29 But in those use cases you cannot just say Oh, an exception has been thrown.

15:02:33 I don't care anymore about like the the runtime of this function being being non-deterministic.

15:02:37 You just can't do that, and therefore you just end up not using exceptions at all right.

15:02:43 And I think, Patrice actually at one of the earlier earlier



meetings actually mentioned that you know there is like a per potential performance overhead, with just exception handling kind of even if you don't even call that the

15:02:59 exception. he was doing some sort of benchmarking, I believe.

15:03:03 This was, I think, like, you know, very, very long time ago.

15:03:07 But it it is kind of it should be kind of a note noted, I think, that you know it's there, or it is known.

15:03:16 I I think that Sutter was actually mentioning like making it, you know, kind of turning the exception handling P path to be more deterministic in order to or kind of in the return type to basically not necessarily

15:03:33 dynamically allocate when you're doing an exception when you're generating, and string that you're kind of doing the the exception that was probably one of the things that actually hurts with the exception

15:03:45 handling. Yes, so So there are 2 different things here. one is what happens when you throw an exception, and what happens when you don't throw an exception.

15:03:52 If you throw an exception currently that's a dynamic allocation which is not a deterministic and that's just the way like the language works.

15:04:04 You cannot really do anything about that. So her was addressing that problem with his proposal Right?

15:04:09 He was trying to find a deterministic way. of like throwing catching exceptions in a different way that doesn't require rtti and memory allocations.

15:04:18 So that's one problem by the way. i'm curious you know what happened to this proposal.

15:04:22 I think they have to know developments there in the last 3 years since this was released.

15:04:27 But if anybody knows anything else i'm i'm curious but then the other thing is once you finish all initial discussion. Yes, and so the other thing is what happens.

15:04:37 If you have a try, catch block there in your code.

15:04:41 And you have exceptions enabled but you don't actually throw an exception.

15:04:45 Does that have any overhead, and the heck and benchmarks there?

15:04:48 I think the number of people did that. I think Ben Craig also had a paper where he was looking at this

15:04:55 It's kind of subtle so but at the end of the day it boils down to

15:04:58 You have, like 2 possible strategies how to implement exceptions, right?

15:05:02 Because you need to store all this information about how to unwind the stack that that needs to go somewhere.

15:05:08 Right. So either you generate that information at one time and then you get runtime overhead, which is what the windows 30 two-bit does,

or you generate that information at compile time, and you store it somewhere in your binary and that's

15:05:20 what windows? 60 four-bit does That's what Linux does.

15:05:25 That's what like android does so you don't notionally have runtime overhead.

15:05:31 However, you have you have more stuff in your binary so that's gonna affect code layout.

15:05:36 So you can still indirectly effect performance anyway.

15:05:40 So, but it's kind of very hard to measure and it depends on how exactly you set up your benchmark, and so that kind of stuff.

15:05:47 So that's how far my knowledge on this topic goes if if anybody knows more about this.

15:05:52 I would be very happy like very, very curious, but let's first hear a guy his hands Do you want to go first?

15:06:03 Oh, I thought I thought you wanted to reply to this but well, I wouldn't just definitely find something else.

15:06:10 Okay, that's just regarding habits exceptions it's it's a great paper.

15:06:17 But you know. he's one man who is short on time and it's been devoting all of his energy to secretly, the next the new syntax called about developing a new syntax as far as

15:06:29 i'm understand he's expecting to wind in some of these other stuff that you know is Meticlasses proposals, and is exception proposals into the new syntax rather than channeling them into the

15:06:44 into the into c What are the problems with the exceptions since there's an awful lot of time being invested.

15:06:54 Buy some companies into making care the exception safe where it's not necessarily the case.

15:06:57 The form you know they're they're operating on a different strains, So that has been pushed back against her static exceptions.

15:07:04 Paper. So so basically it's it's not going anywhere is that basically the executive summary Oh, rather pessimistic one.

15:07:15 Yeah, okay, Well, that's good to know at least get left

15:07:22 I've also personally said, Well, if you saw an exception you don't care about your real time deadline, and with that I mean well, I do not in safety.

15:07:37 So the point is you have to go to your safe fall back before you scroll your exception, and that is now the reason why you don't care about your deadline anymore.

15:07:54 Because you do everything that you need to do before your deadline before you scroll your exception.

15:08:04 Yeah, So I think it really depends on the use case. I think I can see, use cases where that's the case.

15:08:10 I can also see use cases. where you're like on some kind of regular callback that you can't just stop right.

15:08:21 And so you just need to, simply, which is the dummy.

15:08:22 Call that more or less, that not on the dummy a safe call back that works like you fading out, or whatever you have to do that first, and then you go for the exception.

15:08:39 Yeah, that means, Michael. Yeah, So quite a few quite a few streams or thoughts there. i'm i'm highly interested in this, because i'm I care both about the high performance side and the safety side they're almost like opposite ends

15:08:58 of of the spectrum, sometimes not all the time. Sometimes they do coincide.

15:09:02 So yeah, too. dead love. Yeah, I get it that's a pretty good technique standard technique where you do all the real time stuff, right?

15:09:10 And then you, and then you throw the exception. So anything that requires a real-time response. You get that done, and then you you

15:09:17 You put you then you go to the part that you don't care about with the deadline is no longer important.

15:09:24 The the hoop paper. I have talked to her to see what his intentions are, because I care about it in some ways not exactly in this current form.

15:09:35 Because there's an th the it has an abi incompatibility, because it causes a different another parameter like the Vv. table parameter that's hidden to be added to your to your function

15:09:48 call and so in that way it's not it's not yet ideal.

15:09:54 But as far as I understand it, hopes held it back.

15:09:58 In favor of trying to let's get c 23 through first, and then he might come back to it.

15:10:04 Because this is obviously a big discussion about about exceptions and how to handle it.

15:10:09 And of course, in this group we've been shepherding through a paper from a gentleman at

15:10:15 Is he at all now, or something like that? What he did his PHD.

15:10:20 Thesis on deterministic exceptions for embedded systems. and

15:10:24 His name is Jeremy Renwick. i'm sure you can easily Google his paper.

15:10:28 And anyway, we've reviewed this in this group 2 or 3 times now trying to find a way building his system of compile time exceptions.

15:10:39 That is deterministic for embedded systems that works with embedded systems.

15:10:45 Into the C Standard and we've not figured out a way at which, so far, it's a great experience paper.

15:10:53 The fact of it is that we don't know what it takes to change the standard, and to to to make that possible.

15:11:04 Even now, I mean, we know that with the c standard exceptions doesn't have to be built exception.

15:11:11 Information does not have to be built on the on the heap.

15:11:13 It could actually be built on the stack the standard Doesn't Doesn't say it has to be built on the Hebrew.

15:11:19 It just says it has to be built somewhere it's just that all compiler vendors have used a heap to build that exception.

15:11:24 Information that's what causes the non-determinism, the memory allocations right?

15:11:30 The dynamic memories. and this is why but it doesn't have to be, you know i'm i'm.

15:11:36 I actually wrote an exception system for Ibm compiler is, an I put it on the heap just because the directions from that time you know, if you have big iron machines, space is not really a matter of contention so you could pretty

15:11:53 much use as much space as you want once exception is thrown and the's what they do that's what they do Once the exception is on you just start gobbling up space to store all that information and that's what

15:12:03 causes, the the slowdown, the unwinding, the personality routines all takes time.

15:12:08 And this is what this is. This is why all the exception systems that we have caters do that kind of that kind of big iron, big mainframe systems.

15:12:17 Because they have lots of memory they don't cater to embedded systems where memory is limited because that's not what our bosses told us to do.

15:12:24 But now it's different. it's getting to the point where you do have limited resource limited memory, and you do want to do a an exception system that conforms to that that means putting it on the stack and the

15:12:37 problem, and that's okay, and both herbs paper and Jeremy Reynolds paper essentially do that they try to put the exception stack frames and the the exception frame information on the Stack and this is

15:12:52 why they can be much more deterministic they're proven to be deterministic by data, because they've done benchmarks and all that on on these kinds of things.

15:13:00 The problem is, no one has built it, even though these things, this paid these papers at that PHD.

15:13:06 Thesis has been out for 3, or 4 years. now no one has built it, and I don't anticipate people building it for another 3 to 5 years. so I just don't see it as being an active solution, even though in

15:13:18 theory, it should all work and multiple people have proven it.

15:13:23 So are you basically saying that this, the problem is not in the specification.

15:13:28 But the problem is in the kind of compile implementation.

15:13:31 The problem is exactly what i'm saying I don't believe that interesting Yeah, the specification specification the C standard does not prohibit you by by by to implement exception on the stack.

15:13:42 I can point to the exact paragraph I gave and talks about this in

details. Say you know there's nothing here that says you have to put it on the heap.

15:13:49 It's just that by convention everyone has put it on the heap.

15:13:53 So, Guy, I'll call you in a second but I just wanna reply to this doesn't that or maybe yeah guy go first.

15:13:59 Maybe it's gonna say actually in discussion with her about the exceptions.

15:14:05 Paper. the issue of running out of memory has been quite important. Because at the moment, one problem with throwing exceptions is, if you're throwing an exception, because you run as memory why?

15:14:16 Do you put the exception but it was observed that the whole out of memory. Business is pretty much meaningless now, because memory allocation, It's not simply a case.

15:14:30 You know it's it's often it's simply a case of you know I like you know marking a page is available for being written to, or something like that.

15:14:36 I mean you actually run out of memory, or you observeably run out of memory long after you've made the allocation.

15:14:43 So the actual classifications of situations in which you can throw exceptions, have been diminishing.

15:14:50 Putting exceptions on, you know the the opening exceptions on the heat thing.

15:14:55 I I think it's becoming a it's it's sort of a non issue, because memory just doesn't work in the same way that it works in the fine to you in the 1980 s when all this was when all this was

15:15:07 pairing. Wait, you mean running out of memory is not an issue, because obviously obviously putting an exception on the heap, can still, you know, actually result in a in a memory location that you can where you can observe the the issue is that what

15:15:23 what when a memory allocation is made then for example if if you like, you can request medication from of anybody's direct experience windows.

15:15:35 But I believe the same is true with in the Linux.

15:15:37 You then the allocation doesn't fail until you actually try and write the memory. but by by which point it's too, it's it's too late you don't have to the failure happens doesn't happen at

15:15:49 the point of allocation, it happens to the point of use.

15:15:52 So. yeah, what you want to do is show the exception at the point of allocation. Not at the point.

15:15:56 And That's that's not necessarily something that that can be done anymore.

15:15:59 But like, from what I understand, like herb saying that if you run out of memory, that shouldn't be treated as an exception, like basically like bad Alloc, or something, it should be treated as you run out of resources behavior is

15:16:15 undefined, and like the whole thing. So I think I think that actually that is actually very reasonable approach.

15:16:24 But I I'm very curious about like I wanna make a comment about one other thing that Michael said earlier, because I think that's really fascinating.

15:16:31 It's not a perspective, I have really heard before that the kind of issue of non deterministic, exceptions, and I'm not talking about running out of memory, which is exceptions in general.

15:16:45 The the issue of non-deterministic exceptions is not an issue of language specification.

15:16:50 It is an issue of tooling and compiler technology.

15:16:53 And so that makes a lot of sense to me I just haven't heard this before, And so I wonder then what the point is off things like hubs.

15:17:02 Okay, if if the problem is in the implementation of an exception mechanism rather than in its specification, if we could theoretically do this today, the only point is that the only point of his paper that we have a different syntax which

15:17:16 is distinct from the existing mechanism, basically for backwards.

15:17:21 Compatibility purposes. is that like the only interesting thing?

15:17:24 Yeah, Well, there, there's also. the abi break right but but there's not the point of the paper is to switch people's mindset from things that.

15:17:33 It has to be only be implemented on this on the heap, and now it could also be implemented on the stack.

15:17:39 But this putting on a stack causes another, another another kind of problem, which is that the Abi is going to match your previous calling.

15:17:47 Ap Api. Yeah. So if you do care about backwards compatibility and still supporting the old exception mechanism at the same time, Yeah, and like the Abi, then you need like 2 parallel

15:18:01 mechanisms, which is what Herb has done there, and his proposal.

15:18:04 But you. don't strictly need that if you only care about the non-determinism.

15:18:09 You don't care about backwards compatibility you don't need a new syntax for any of this.

15:18:13 Is that my correct like t of understanding but you don't actually need a new syntax there, and you could just implement that says, Oh, i'm gonna now compile it using a stack mechanism fine go ahead

15:18:31 fascinating, i'm not going to agree with any other library. Okay, fine. just go nuts. Well, it feels that to me that maybe the who need it, and who do things anyway like compiling with f no exceptions, or having their own

15:18:50 if St. L. or whatever. right, then you might as well give them another compiler flag, and then it least like the compiler is gonna take care of this stuff for you.

15:18:59 You don't have to do it yourself. so it feels like that would be a

great solution, actually.

15:19:03 So I hope maybe somebody's gonna somebody's gonna do that Anyway, debt and I've had signs up

15:19:12 Actually, I believe, Gcc. has some kind of this compile of like could be, because Spec.

15:19:23 In 2,001. I think it was when we heard from Ibm the first time about this table based approach.

15:19:34 At that time everybody was using the tech approach and only afterwards.

15:19:42 It got into the Itadium Abi And and that is the real problem.

15:19:53 Because, I can remember, I think it was 2,005 or 2,006 at the beginning of Wt.

15:20:02 21 meeting how a tenant from apple at that time.

15:20:12 Speech with an official statement from Apple and Photoshop.

15:20:21 If the exception, Abi we'll change they will do strongly against any standard version that contains that one, because that would mean that the plugins for Photoshop don't work anymore.

15:20:44 For example, and all the the the exception api if you don't have a pure C interface between 2 components.

15:20:58 The exception Api is a very important part of that interface.

15:21:07 And since this is so, you really need to keep that from the compiler. point of view, maybe because that is what your customers want, and that is only 2 for the desktop systems.

15:21:26 As soon as you are on embedded systems where you compile everything you yourself, anyway, in most times you don't care.

15:21:37 Yeah. for example, if you have queues Steinba queue is all the plugins very go through the exception.

15:21:46 Interface.

15:21:50 Guy that's absolutely fascinating delt I shall remember that.

15:22:00 I was going to suggest that we might have great ideas about well, that's just right.

15:22:06 A stack-based exception, Adelaide.

15:22:08 But there's a you know there's a month. you know there's a cost to doing all these things.

15:22:15 This is possibly going to be too great for our compiler vendors to

15:22:17 That I think with the possible exception of Gcc.

15:22:22 And clang unless people actually start saying that we want a stack based exception outlet.

15:22:29 Then we're we're not going to get one in the General case, unless somebody is prepared to write off the clang in Gcc.

15:22:37 And you know, make it obvious, and advertisers extensively one of the problems we've always had with talking about exceptions simply measuring the wretched things it's been.

15:22:48 It's always been very hard to compare exception safe and exception unsafe code, because one makes different assumptions.

15:22:54 Fundamentally all through your code base, based on the decision of whether or not you're going to use exceptions

15:23:05 Yeah, but I think it's a good point that you know we don't know whether that's realistic expectation from kind of compiler vendors that they provide basic exception. so whether it's it's too costly or whether there is

15:23:19 a market for it because it's it's weird like i've been in the low agency business now for over a decade, and I have never heard about this before.

15:23:27 So I wonder if it's just me being completely ignorant or whether it's just not as widely known as it should be that you know that is something that's technically possible.

15:23:38 And that's the reason why there is no demand for it currently

15:23:42 Yeah, if I may. you you're probably not that different than most of us. I myself really only learned of this in the last 4 or 5 years after I looked at the paper from her I looked at this Guy's.

15:23:55 Phd. pieces. It became clear to me that this was the truth.

15:23:57 I imagine not a lot of people know about this and because it's one of those paragraph in the standard that is pretty well hidden, and but there's no there's lots of paragraph in the standard

15:24:08 as well hidden, unless unless we actually, unless we actually wrote it.

15:24:12 Most of us gonna struggle to find it. But yeah, you know the the thing is that I I mean, I I suppose.

15:24:23 Yeah, I I suppose that the the thing that I guess I want to transfer across is I've been looking at this for a while, trying to figure out what to do with with low latency and I pretty much feel like I I know

15:24:35 we all a lot of the problems are we guys we've already talked a lot about dynamic memory.

15:24:39 Well, people are using tool allocators static a whole static chunk to make sure that it's deterministic.

15:24:47 Okay, so that's kind of solvable kind of solvable with exceptions, because it's so infused in the Standard Library.

15:24:53 It's hard to solve, because it's not just about not using the exception.

15:24:58 I do use. use a whole different exception. system that stack base, or somehow you just bracket out the exception, which is why i'm very curious about the solutions you guys talk about i'm mostly coming from this from a safety

15:25:09 self-driving call point of view. and yeah, you have to have your own Stl: Okay, that's not great, but doable.

15:25:15 I guess I can somehow make my own sdl that doesn't use exceptions.

15:25:23 That's not that's happened quite that's happened a lot before, you know. ea electronic odd had their own Stl.

15:25:29 And I imagine that most of it doesn't use any exceptions and just



use error code or something like that.

15:25:34 The electronic the electronic Sdl: you devoted the exception throwing problem by simply not implementing things that could error which was which was awkward.

15:25:47 The main reason why the Estl existed was to deal with memory, allocation, and fragmentation.

15:25:53 Oh, okay. So they didn't deal with the exception at all. Okay, it's good.

15:25:57 So i've seen kind of 2 approaches one is you just artificially restrict your you compile with exceptions, but you just artificially restrict yourself to the subset of Stl.

15:26:10 That that can never throw any right and Then you're right kind of your own facilities on top, and you don't use exceptions, or you just throw the whole scl out the window you write your own and then you

15:26:21 compile with no exceptions, and as an error mechanism, you use either return codes, or you use something like stood expected, which, you know is now coming in C 3.

15:26:33 But I think pretty much every like application framework under the sun has something similar already.

15:26:39 It's just that now. we're gonna get it as a vocabulary type which makes it useful across api boundaries, which is another very interesting topic.

15:26:45 I'm not going to go into now. but yeah this is kind of what I've seen people do in this space.

15:26:53 And I don't know anything about automotive for finance.

15:26:54 I'm talking about games audio. you know these kinds of kind of consumer, like thing like low latency things that kind of run consumer software on consumer hardware.

15:27:06 Roughly speaking, using a off the shelf kind of operating system like windows.

15:27:13 Mac Linux, rather than, you know, something like their metal or the real time operating system, or something.

15:27:19 Guy. So one final point, I guess, on the whole, rewriting the sdl business.

15:27:26 I think we're writing sdl is a really bad idea, because the problem that you're solving is that the containers that the str provides are not up or not containers that you want to use

15:27:37 rewriting vectors that it doesn't throw exceptions means that you don't have a vector anymore.

15:27:41 You've got something else that looks like a vector that actually It's still a vector as we understand it's in the standard.

15:27:48 And you know we have an object called a dynamic array, which but he's like back to. but it doesn't throw exceptions, and that's what we use in the general case instead.

15:28:03 I think you know, rewriting the St. I was throwing the baby out

with the bathwater.

15:28:07 The Stl does carefully describe what the expected behavior is in containers, and that behavior can be met without throwing exceptions.

15:28:15 You know, the all all the you know the the type tests that or the the typewriter is that the container should have, don't imply exception handling, So it's just it's it's quite used to right around

15:28:27 containers that that operate with the rest of the you know, with the algorithm, for example.

15:28:33 So that is true. But for example, in Audio, we have another constraint, which is, we can never ever make a dynamic memory allocation in the real-time path, and Right? you're in consent.

15:28:44 So then, yeah, you write our own container which Isn't. A. vector which is something like static, vector which has a static capacity where everything is like inside the object.

15:28:52 And then the Ab. the Api changes then you can't have a pushback anymore.

15:28:56 You have to have a try push back which can fail right and stuff like that.

15:28:59 So so. yeah, I guess you're right like either you you can meet your needs with like the existing api or you can't, and that's kind of a decision you have to make at that point

15:29:15 That was good information. I guess I guess my summary Is that okay?

15:29:18 So you can't rely on this static mechanism coming anytime soon.

15:29:22 I mean it works but it's nobody nobody's implemented It'll probably take somebody 3 or 5 years to do it. You can You can marginally rely on the idea that you can bracket yourself away from the stuff that 15:29:32 throws exception, which is kind of what I think Guys trying to point me to.

15:29:36 And I agree. Rewriting The Sel is just growing the whole baby out with the bathwater. not good.

15:29:40 So what what else? What choices do people have left Now at this point You still have to.

15:29:45 If you're gonna use the standard library, and assuming that's that's a fundamental requirement, because you can always write your own application and don't use exceptions.

15:29:51 But that's not that's Not the world We live in especially in the commercial gaming.

15:29:55 I put hft world people will have to find a way to Marshall, sometimes using error codes in the time.

15:30:04 Critical path, and sometimes using exceptions in the one in a 1,000 case, time, non critical path, exceptions, as I understand, it, was never supposed to be used in in any kind of hard, or even soft, real-time requirements.

15:30:21 That was never be honest design. It was designed for the one in a 1,000 case.

15:30:24 That's that an error happens so most errors are not one in a 1,000.

15:30:28 They're one in a 100 or one in 10 so i'm just thinking that. is there a way that we can you know.

15:30:33 Do we? you know well like, like, exceptions like sorry like like variants and optional where somehow you can take an exception, and somehow it's maybe you can still pass out an error code if you need it so they can keep

15:30:49 the software running. I don't know like something like that I think we have to have an an immediate like an like a current best practice solution for people, because we know that the other great solutions are not gonna land anytime soon So I

15:31:02 think I have one possible answer to that. But Guy had his end first.

15:31:06 Sure I wanted to point out that exceptions solve the problem of unexpected things happening.

15:31:16 Now in the game, domain the it's the kinds of inputs that the game has tremendously limited, you know, tremendously limited.

15:31:29 Basically those that the number deterministic inputs you'll get into a game is from a controller.

15:31:34 Maybe from a file system and that's it so actually throwing exceptions.

15:31:41 It's kind of outside of some of the expected scope of gain developments, anyway.

15:31:47 This doesn't answer your question, Michael, but I just wanted but I just wanted to make clear that you know you're using exceptions.

15:31:54 Need to be a a for exceptional cases from outside of your system.

15:31:57 Okay, that that's good data. Yeah, yeah I mean for a call. We have to kind of figure out all the possible it exception cases.

15:32:05 But but yeah, no, I I understand. Yeah, so another approach that, I think is really cool, is, if you use something like stood expected which we now have in understand that you can actually build a lot of things, like especially since it's a

15:32:22 vocabulary type that you can pass across like api boundaries, and, like different libraries, can agree on on that being kind of the expected unexpected type that we now use.

15:32:31 There's like really cool patterns, you can do and i've shown some of them in my cbp cone keynote a few weeks ago, where you know, you can kind of pass pass expected across interfaces you can use them

15:32:45 in algorithms where, like, you can make them generic on the type of error that the expected contains.

15:32:52 You can have these like things. We have an expected of a variant of different errors.

15:32:59 And then you! you kind of have different errors coming in from different layers, and, like you kind of collate them where you do your error handling, and you can do is to visit on the variant, and then the if

that looks very much like a

15:33:08 try catch block, except it actually forces you to like catch all the different error types.

15:33:12 Otherwise you get a compiler error so that's like really really interesting.

15:33:17 Kind of design patterns there, or like coding patterns there, that give you a lot of the functionality of expect exceptions, accept like the deterministic and efficient with the downside that there was just a lot of syntactic overhead, right? because it's a library type.

15:33:35 Yeah, you have to like explicitly construct these error types and stuff like that.

15:33:38 So it's not syntactically as neat as exceptions are, and it's and there's no like separate kind of mechanism like you can throw out of a function. you can still only leave a function through a return statement right

15:33:49 so there's a lot more syntactic noise.

15:33:53 But you get effectively like a very, very similar behavior, and sometimes even better stuff, like compile time checked like you.

15:34:00 You caught all the cases and things like that, and you can.

15:34:02 You can do that with it. expected quite, quite nicely. Thank you.

15:34:09 So i'm kind of a bit conscious of time you have 25 min left, and wondering if there was anything else on the agenda that people want us to discuss today, or if not, we can just like let this discussion continue until you run

15:34:23 out of time. But yeah, that on the agenda they were like a few papers here which I don't think we have the authors in for

15:34:34 There. there was meant to be a discussion about games, topics. No, those those are for different week, different months.

15:34:41 Every month I alternate between different topics like games and and

15:34:48 Okay, so none of these are relevant for today.

15:34:54 Okay, that's great. Then we can just keep this discussion going for another 20 min, and then we can stop. If people If people have had enough of us.

15:35:02 Yeah, let's just see if there are any more hands I think it's a very fascinating discussion.

15:35:07 By the way, I i'm learning a lot here and I will definitely look at that transcript later.

15:35:13 As well. so thanks for everybody who's but Yeah, no thanks, thank you.

15:35:19 Thank you timer for volunteering you've done you've done a great job.

15:35:21 This is fantastic. i'm actually back but I have to leave again soon.

15:35:27 So i'm happy If we stop here. so and then, that way. I can just save the transcript right doesn't. Oh, can I have a ask you a question?

15:35:35 Are you gonna publish the transcript somewhere? Is that going to be accessible?

15:35:38 Oh, oh, oh, because this is not a face to face. I I emailed the entire transcript to the reflector.

15:35:45 : Yeah. So yeah, okay, Cool: alright. So is there any more discussion on this?

15:35:55 Otherwise we can can wrap up alright. Thank you. Thanks.

15:36:02 Everybody. yeah, normally next month, so the next. meeting is going to be on December the seventh.

15:36:14 And it will be game I believe it's also a Wednesday.

15:36:17 It's always wednesday isn't it it's also a Wednesday.

15:36:26 It's always wednesdays, isn't it but because time change differently. i'm always just using my own time, and just let everyone else convert to it, whatever their time is.

15:36:36 Yeah. So So maybe we like another approach is to always give the time in Utc, because that's like really unambiguous.

15:36:41 And everybody knows what that means. But yeah I don't know if I had to look at that.

15:36:46 I think Utc moves around, too, because of stay like savings.

15:36:50 Time. yeah time to time just move around so one thing you see, alright.

15:37:00 So then i'll see some of you in kona and then i'll see all of you hopefully on December the seventh.

15:37:09 And Thank you very much for this discussion. cheers guys, Goodbye.

On Wed, Oct 12, 2022 at 12:13 PM Patrice Roy <patricer\_at\_[hidden]> wrote:

> I'll be in class today during the meeting so I cannot make it, sadly :(

>

> Le mar. 11 oct. 2022 à 23:31, Michael Wong via SG14 <sg14\_at\_[hidden]>

> a écrit :

>

>> Topic: SG14 Low Latency Monthly This meeting is focused on Low Latency.

>> There were several Low latency discussions on the reflector this month and

>> this would be a good time to review and summarize to see if a paper can be

>> jointly published. Alternatively, we can continue with the Games paper that

>> was started at CPPCON.

>>

>>

>> Hi,

>>

>> Michael Wong is inviting you to a scheduled Zoom meeting.

>>  
>> Topic: SG14 monthly  
>> Time: 2nd Wednesdays 02:00 PM Eastern Time (US and Canada)  
>> Every month on the Second Wed,  
>>  
>> Join from PC, Mac, Linux, iOS or Android:  
>> <https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJuT1loeXpKbTcydz09>  
>> Password: 789626  
>>  
>> Or iPhone one-tap :  
>> US: +12532158782,,93151864365# or +13017158592,,93151864365#  
>> Or Telephone:  
>> Dial(for higher quality, dial a number based on your current  
>> location):  
>> US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1  
>> 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833  
>> or 877 853 5247 (Toll Free)  
>> Meeting ID: 931 5186 4365  
>> Password: 789626  
>> International numbers available: <https://iso.zoom.us/u/abRrVivZoD>  
>>  
>> Or Skype for Business (Lync):  
>> <https://iso.zoom.us/skype/93151864365>  
>>  
>> 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%3Dl%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

>>

>> 1.2 Adopt agenda

>>

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

>> previously approved minutes to ISOCPP.org

>>

>> 1.4 Action items from previous meetings

>>

>> 2. Main issues (125 min)

>>

>> 2.1 General logistics

>>

>> CPPCON minutes:

>> <https://wiki.edg.com/bin/view/Wg21virtual2022-07/SG14>

>>

>> Future meeting plans

>>

>> \*No call Nov due to Kona F2F:

>> \*Dec 7, 2022 02:00 PM ET Games

>> \*Jan 11, 2022 02:00 PM ET: Embedded

>> \*Feb 8, 2022 02:00 PM ET: Finance/low Latency

>> \*Mar 8, 2022 02:00 PM ET: Games

>>

>> 2.2 Paper reviews

>> Discussion on Embedded:  
>> Review latest mailings:  
>> P2532 Removing exception\_ptr from the receivers concept  
>> Based on the last meeting and the discussions here.  
>> P2544 C++ Exceptions are becoming more and more problematic  
>> We might want to chime in here.  
>> /Paul  
>> P. S. P2327 de-deprecating volatile received a "consensus" straw poll.  
>>  
>>  
>> Discussion on Low Latency/Finance topics  
>>  
>> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p1839r4.pdf>  
>>  
>> Patrice's paper on games.  
>>  
>> P2300  
>> Swift  
>>  
>>  
>>  
>> Discussion about Games topics:  
>>  
>> P2388R1 - Minimum Contract Support: either Ignore or Check\_and\_abort  
>> <<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>>  
>>  
>> Patrice's WIP on Games issues.  
>>  
>> Finance topics from July 14, 2021.  
>>  
>> <https://lists.isocpp.org/sg14/2021/06/0636.php>  
>>  
>> <https://lists.isocpp.org/sg14/2021/07/0642.php>  
>>  
>> 2.2.1 any other proposal for reviews?  
>>  
>> Deterministic Exception for Embedded by James Renwick  
>>



>> [https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low\\_cost\\_deterministic\\_C\\_exceptions\\_for\\_embedded\\_systems.pdf](https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf)

>>

>> Freestanding?

>>

>> 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: 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  
>>  
>>  
>> \*No call Nov due to Kona F2F:  
>> \*Dec 7, 2022 02:00 PM ET Games  
>> \*Jan 11, 2022 02:00 PM ET: Embedded  
>> \*Feb 8, 2022 02:00 PM ET: Finance/low Latency  
>> \*Mar 8, 2022 02:00 PM ET: Games  
>>  
>> \_\_\_\_\_  
>> SG14 mailing list  
>> SG14\_at\_[hidden]  
>> <https://lists.isocpp.org/mailman/listinfo.cgi/sg14>  
>>  
>

