# C++ Ecosystem Evolution Meeting Agenda

## Quick Start

Go to Agenda. That is the tentative agenda for the next meeting.

## Table of Contents



## About

This is the agenda document for the development of the Ecosystem Evolution group, which is forming to launch the CPS project.

This document is an open agenda. Anyone attending the meeting is allowed to suggest items for the agenda by adding bullet points to the Agenda. It is recommended that you add your name to the agenda item so that the attendees of the next meeting know whose concern is being discussed.

Attendees of that meeting are allowed to adjust the agenda during the meeting, including deferring agenda items to the future or removing them from the agenda. Actual discussions will be recorded in the C++ Ecosystem Evolution Meeting Minutes and then shared on the C++ Ecosystem Evolution mailing list.

## Meeting Purpose

It is expected that success for C++ (and polyglot!) packaging standards will require much more efficient processes than a periodic synchronous meeting. Most work will need to be done asynchronously via usual open-source best practices (issues, release managers, day-to-day decision-making by moderators and maintainers, etc.).

That being said, it is also expected that bootstrapping a community, providing momentum for the project, and working out larger cultural and teamwork issues will benefit from interactive, face-to-face discussions. This meeting exists to serve that purpose.

## Staying in the Loop

Organization of this meeting, including finding times to meet and disseminating minutes, will be done over the C++ Ecosystem Evolution mailing list. Subscribe to that list if you would like to be included in the C++ Ecosystem Evolution community.

## Automated Meeting Minutes

Meetings generate transcripts, which we plan to use to create meeting minutes. If you are uncomfortable with this, please address it at the beginning of the meeting.

## Adding to the Agenda

This document is publicly commentable. Since it will eventually be publicly shared, it is not publicly editable in order to stay ahead of eventual bad actor problems. If you would like to be able to edit the document, please ask for those privileges via the Google Documents API or otherwise contact mail@bretbrownjr.com.

## Agenda

The following are items to discuss for the next meeting, sorted roughly in order of priority. Please add your name so the group can ensure you are adequately included in the discussion. Items may be deferred to a future meeting by adding text like “(Defer to Jan. 2038)”.

- Introductions (new folks joining us this week)
- Impact of Contract Assertions on Eco-system
- MW/Vito: Last call re: proposed CPS schema change to support modules (#95)
- MW: Please *use* the CPS JSON schema; thanks!
- Philippe: Intro and chat about Package-URLs and SBOMs with CPS
- Bret: Revisit group organization?

### Recurring

- [Kitware / CMake] CppCon Sept 13-19 2025
    - full integration of CPS into CMake
    - move to marketing mode to a wider audience
    - pkg-config conversion/usage figured out
    - C++ 20 module support

### Backlog

The following items need to be discussed at some point in the future, but not yet.

- Project governance
- CPS repo docs
- Making CPS approachable for newcomers
- Code of conduct

## References

- Meeting Minutes: https://docs.google.com/document/d/1NR4zE4GqLdVhUzthNg__tN43hRYGXUxyqOymMvujvQY/edit
- C++ Ecosystem Evolution mailing list
- CPS Repo: https://github.com/cps-org/cps
- CppCon 2023 CPS Talk: https://youtu.be/IwuBZpLUq8Q

## List of Ecosystem Evolution-related papers

2018, P1040R0, std::embed, SG15, Evolution, Library Evolution

2018, P1052R0, Modules, Macros, and Build Systems, SG15, Evolution

2018, P1067R0, C++ Dependency Management: Package Consumption vs Development, SG15

2018, P1177R1, Package Ecosystem Plan, SG15

2018, P1178R0, C++ Compile, SG15, Library Evolution

2018, P1204R0, Canonical Project Structure, SG15

2018, P1254R0, Notes on C++ Package Management, SG15

2018, P1275R0, Desert Sessions: Improving hostile environment interactions, SG15, SG16, Library Evolution

2018, P1281R0, Feature Presentation, SG15

2018, P1300R0, Remember the FORTRAN, SG15, Evolution

2018, P1313R0, Let's Talk About Package Specification, SG15

2019, P1482R0, Modules Feedback, SG15, Evolution

2019, P1484R1, A uniform and predefined mapping from modules to filenames, SG15

2019, P1634R0, Naming guidelines for modules, SG15

2019, P1687R1, Summary of the Tooling Study Group's Modules Ecosystem Technical Report Telecons, SG2, SG15, Evolution

2019, P1688R0, Towards a C++ Ecosystem Technical Report, SG15, Evolution

2019, P1767R0, Packaging C++ Modules, SG15

2019, P1788R3, Reuse of the built modules (BMI), SG15, Core

2019, P1832R0, Improving Debug Builds Inline With User Expectation, SG2, SG14, SG15

2019, P1842R0, Generalized Module (Dependency?) Mapper, SG15

2019, P1845R0, 2019-09-21 Denver Tooling Meeting, SG15

2019, P1876R0, All The Module Names, SG15

2019, P1905R0, In-Source Mechanism to Identify Importable Headers, SG15, Evolution

2019, P1948R0, Modules: Keep the dot, SG2, SG15, Evolution

2020, P1184R2, A Module Mapper, SG15

2020, P1838R0, Modules User-Facing Lexicon and File Extensions, SG15

2020, P1857R3, Modules Dependency Discovery, SG2, SG15, Core

2020, P1864R0, Defining Target Tuplets, SG15

2020, P2073R0, Debugging C++ coroutines, SG15

2020, P2074R0, Asynchronous callstacks & coroutines, SG15

2021, P2409R0, Requirements for Usage of C++ Modules at Bloomberg, SG15

2021, P2473R1, Distributing C++ Module Libraries, SG15

2022, P1689R5, Format for describing dependencies of source files, SG15, SG16

2022, P2429R0, Concepts Error Messages for Humans, SG15

2022, P2514R0, std::breakpoint, SG15, Library Evolution

2022, P2515R0, std::is_debugger_present, SG15, Library Evolution

2022, P2536R0, Distributing C++ Module Libraries with dependencies json files., SG15

2022, P2546R0, Debugging Support, SG15, Library Evolution

2022, P2565R0, Supporting User-Defined Attributes, SG15

2022, P2577R2, C++ Modules Discovery in Prebuilt Library Releases, SG15

2022, P2581R2, Specifying the Interoperability of Built Module Interface Files, SG15

2022, P2673R0, Common Description Format for C++ Libraries and Packages, SG15

2022, P2701R0, Translating Linker Input Files to Module Metadata Files, SG15

2022, P2702R0, Specifying Importable Headers, SG15

2023, P2656R1, C++ Ecosystem International Standard, SG15

2023, P2717R5, Tool Introspection, SG15 Tooling

2023, P2791R0, mandate concepts for new features, SG15 Tooling

2023, P2800R0, Dependency flag soup needs some fiber, SG15 Tooling

2023, P2803R0, std::simd Intro slides, SG15 Tooling

2023, P2898R1, Build System Requirements for Importable Headers, SG15 Tooling

2023, P2962R0, Communicating the Baseline Compile Command for C++ Modules support, SG15 Tooling

2023, P2978R0, A New Approach For Compiling C++, SG15 Tooling

2023, P2990R0, C++ Modules Roadmap, SG15 Tooling

2023, P3033R0, Should we import function bodies to get the better optimizations?, SG15 Tooling

2023, P3034R0, Module Declarations Shouldn't be Macros, SG15 Tooling,Evolution

2023, P3041R0, Transitioning from "#include" World to Modules, SG15 Tooling

2023, P3057R0, Two finer-grained compilation model for named modules, SG15 Tooling

2024, P2977R2, Build database files, SG15 Tooling

2024, P3051R1, Structured Response Files, SG15 Tooling

2024, P3081R0, Core safety Profiles: Specification, adoptability, and impact, SG15 Tooling,SG23 Safety and Security,EWG Evolution

2024, P3092R0, Modules ABI requirement, SG15 Tooling,ARG ABI Review Group

2024, P3267R1, Approaches to C++ Contracts, SG15 Tooling,SG21 Contracts

2024, P3286R0, Module Metadata Format for Distribution with Pre-Built Libraries, SG15 Tooling

2024, P3321R0, Contracts Interaction With Tooling, SG15 Tooling,SG21 Contracts,EWG Evolution

2024, P3335R3, Structured Core Options, SG15 Tooling

2024, P3358R0, SARIF for Structured Diagnostics, SG15 Tooling

2024, P3470R0, Interface-Unit-Only Module Library Support, SG15 Tooling,EWG Evolution

2025, P3696R0, Discovering Header Units via Module Maps