



Fiscal Sponsorship Proposal

Prepared for Boost C++ Libraries

Executive Summary

The [Boost C++ Libraries](#) have delivered substantial benefits to C++ users for nearly three decades, and its needs have changed from the original role in shaping the C++ Standard. To sustain growth over the next thirty years, the project needs new ideas, volunteers, and resources. The proposed Fiscal Sponsorship Agreement (“FSA”) with the C++ Alliance will restore governance rights to the developers while bringing the necessary leadership, expertise, and resources the libraries need to thrive again.

Background

The [Boost C++ Libraries](#) were founded in 1999 by Beman Dawes and others, based on a [proposal for a website](#). The Boost Steering Committee was [formed in 2011](#) to support the project, with the [Software Freedom Conservancy](#) providing [fiscal sponsorship](#). In 2020, Jon Kalb created the [Boost Foundation](#) as a 501(c)(3) non-profit organization registered in Delaware, and then transferred ownership of Boost’s assets held by the Conservancy into the Boost Foundation’s name.

The Boost Foundation organizes the annual [C++Now conference](#), which generates revenue through conference registrations and sponsorships. A portion of these funds, combined with additional donations, is allocated towards supporting the operational expenses of the Boost project. The Foundation owns and administers the [boost.org](#) domain and the “wowbagger” server hosting the website and mailing lists. The Foundation also controls trademarks related to Boost, and a handful of backup domains. In practical terms the Foundation has become Boost’s de-facto fiscal sponsor.

In contrast, the source code, documentation, and supporting tools for the libraries are stored in the [Boost GitHub Organization](#) which is controlled by the developers. Specifically this is defined as the [set of GitHub users](#) which have the Owner role in the Boost GitHub organization, plus their designated proxies. Developers maintain the libraries and publish release versions on a triannual schedule. Traditionally the developers exercise the authority for project-wide decisions, using a consensus-driven approach.

State of Boost

Boost's enormous success is rooted in its two [social technologies](#):

- The [Boost Formal Review](#) is a community-driven process to decide if a proposed library should be accepted into the collection.
- The [Boost Software License](#) strengthens the network effects of remixing source code without burdening forms of compiled code, creating the world's best and most business-friendly license.

Boost's requirement that its libraries use the BSL eliminates the conflicts that can arise from combining code under different licenses. The result is a comfortable and seamless experience for developers, with no ongoing maintenance required to support the license.

Conversely, the Boost Formal Review process demands recurring activity. An author proposes a library, and users offer their endorsements. A review manager is then sourced from qualified volunteers. The review wizard (another volunteer) provides oversight by adding the library to the schedule, supporting the review manager, and resolving disputes. New contributors learn the rules, requirements, and rationale for these roles through [oral tradition](#) and some published documentation.

Formal reviews are both the greatest strength and the greatest vulnerability of Boost. Steady numbers of new library proposals and sufficient numbers of qualified reviewers are needed to maintain the collection's relevance and technical excellence. While the project exploded in activity since its inception, the last ten years have seen consistent declines in participation. A meaningful analysis of the cause is beyond the scope of this project proposal, and also an area of investigation for the C++ Alliance.

Indicators of Boost's vitality such as the number of mailing list posts, the number of reviewers, and the amount of activity in the project's GitHub organization have shown a marked decline. Libraries which previously may have been written and proposed for Boost's formal review, are today likely to appear instead as independent libraries or papers submitted to the C++ Standards Committee.

The project needs a better website, as the current one is more than twenty years old, with a look to match. The wowbagger server needs to be replaced with modern cloud services, as it currently runs on old and unmaintainable versions of the operating system and utilities. There is no official strategy for reversing this stagnation, and no visible top-level interest towards finding a solution.

To endure, Boost needs a new image: exciting, dynamic, and relevant to younger engineers.

Boost Foundation

Boost's current fiscal sponsor has suffered a decline in effectiveness since the loss of its founding members. The subsequent change in leadership has led to mission drift and a shift of focus away from supporting the libraries. This is evident when analyzing their standard of care for Boost:

Infrastructure Support

- The boost.org domain became owned by the Beman Dawes estate in 2020.
- The boost.org domain was at risk of being permanently lost to a domain auction for over four years.
- The boost.org domain registration [lapsed in 2022](#), causing an unprecedented outage of the website and mailing list for over a day.
- The community was not informed of these issues with the boost.org domain.
- The Foundation was unable to source a new, affordable download provider for the release archives when [JFrog](#) discontinued its free service in September, 2023.
- The wowbagger server which hosts the website, mailing list, and release scripts, runs software that is so old that it cannot be easily maintained or upgraded.
- The wowbagger server reports periodic storage failures, and parts of content on boost.org such as library documentation become unavailable.

- There are no published transition plans or resources deployed to ensure service past wowbagger’s end-of-life.
- The problems with wowbagger are acknowledged yet remain unsolved.

Strategic Vision

- Stated efforts to develop strategies for attracting new volunteers to the project have produced no results.
- No original messages posted to the [Boost X account](#) in almost two years.
- The Foundation does not prioritize initiatives to increase the project’s visibility at conferences through promotional materials, and instead strives to minimize its expenses.
- In the June 2024 board meeting with just four members present, the Foundation voted to “cut ties with the Alliance.”

Beman Project

- At the C++Now conference in 2024, the Foundation launched a competing library collection called the [Beman Project](#), describing it as “a new initiative separate from Boost that gets back to its original purpose of improvements to the standard library.”
- The Beman Project was conceived and launched without discussion on the Boost mailing lists.
- The community’s important questions about the Beman Project went unanswered.
- The Beman Project plans to promote itself at the [CppCon 2024 conference](#) using [stickers and shirts](#).
- Beman Project source code and documentation are distributed under the Apache license with LLVM extensions. This complicates their integration into Boost, which uses the BSL.
- Most Foundation board members rarely use the Boost mailing lists, yet are active in the Beman Project’s [Discourse server](#).

- Foundation board members include two active chairs of WG21 working groups, and the Beman Project focuses on the standard library. The Foundation's activities seem to prioritize WG21 involvement rather than Boost.
- The criteria for expanding or replacing the Boost Foundation board is opaque, with recently added members having little to no involvement with Boost.
- Nomination of Foundation board members happens without participation on the Boost mailing list.
- Foundation board members are elected at annual meetings held at the C++Now conference, without participation from the Boost mailing list.

Until recently the community has been largely unaware of these issues, as the mailing list still works, the website is still visible (except for two days when the lapse in domain registration caused an outage), and no one looks too closely at the decline in Boost activity. This year however the community has voiced doubts about the Foundation's effectiveness, calling into question their expansive mission statement and lack of visible action towards improving the project.

The previous arrangement with the Software Freedom Conservancy was governed by a Fiscal Sponsorship Agreement which placed contractual obligations on the parties, including a provision that the Steering Committee determine the technical, artistic, and philanthropic direction of Boost. When the Steering Committee terminated the agreement, the Foundation assumed ownership of Boost's assets without entering into a new agreement between the developers and the non-profit.

The Foundation then exceeded its role by assuming authority traditionally held by developers. This governance overreach has created tension with stakeholders and hindered contributions, ultimately damaging the project's reputation and deterring potential contributors from investing in big, high-risk initiatives. In particular, its obstruction of a modern website that could revitalize the project's online presence.

The C++ Alliance

The [C++ Alliance](#) was formed in 2017 as a California-registered 501(c)(3) non-profit organization dedicated to advancing the interests of all C++ users worldwide. To fulfill this mission, it employs Staff Engineers who develop and publish open-source software under a [permissive license](#). The Alliance has determined that Boost contributions are the most effective means to achieving this.

Executive Team

The Alliance has an all-star executive team working full time to achieve its mission:

Executive Director [Vinnie Falco](#) is a seasoned C++ developer and leader with a proven track record in launching successful for-profit companies and making significant open-source contributions. He's a passionate innovator who enjoys inspiring others with creative energy.

Louis Tatta brings his career expertise spanning finance, development, and management, to his role as Chief Operating Officer. Prior to joining, he was Vice President at J.P. Morgan and worked with the developer behind Bal Harbour Shops, the nation's top-grossing upscale outdoor mall. Louis also held senior roles at Edward Jones and BearShare, where he played a key part in building the company from its inception.

Samuel Darwin, Chief Technical Officer, has many years of experience managing infrastructure, networking, databases, automation, email systems and cloud. Today he focuses that talent on supporting Boost, including the official download distribution, CI systems, and websites.

Board of Directors

The Alliance Board is lean and comprises effective decision makers. As stakeholders in Boost their interests align with the success of the project:

Vinnie Falco, acting president, is a Boost contributor since 2017 whose libraries include [Boost.Beast](#), [Boost.StaticString](#), [Boost.JSON](#), and [Boost.URL](#). His projects in development include [Mr. Docs](#) (a doxygen replacement for C++), [BoostLook](#) (new CSS stylesheets for AsciiDoctor and Antora), and five new networking protocol related libraries to serve as Beast's successor.

René Ferdinand Rivera Morell is a Boost contributor since 2002 whose historical accomplishments include [B2](#) maintenance, and [Boost.Prefdef](#). Previously Documentation Manager, Testing Manager, Release Manager, Owner, [Google Summer of Code](#) mentor, and BoostPro consultant. Currently doing the effort to modularize Boost through B2.

Jon Kalb is active in Boost for twenty five years, who launched and moderated the boost-users mailing list. He grew BoostCon into C++Now, and chaired the conference for a decade. Jon was a founding member of the Boost Steering Committee, and created the Boost Foundation. In addition to serving the C++ community as a freelance trainer and in a number of roles in various conferences and local user groups, Jon founded and continues to serve as chair of [CppCon](#), the world's largest C++ conference.

The Alliance's Board of Directors periodically reviews progress to ensure non-profit compliance and mission alignment, while its full-time executive team focuses on implementing the organization's vision. This allows for timely responsiveness to the projects it serves by avoiding bureaucratic overhead.

Alliance Impact

The C++ Alliance brings resources, expertise, and energy to Boost. Alliance team members are Boost contributors first, staff second. They work independently on their own projects, retain their copyrights, and use personal emails in public spaces. They engage in mailing list discussions, assist other volunteers with their work, and participate in formal reviews.

Over the last seven years, the Alliance activities have made a beneficial impact on the project:

Infrastructure Support

- Alliance deployed [Drone](#) for Boost to provide continuous integration resources after [Travis](#) announced the end of its free service.
- Alliance sponsors the paid [Fastly](#) service which hosts the release archive downloads, in the five months since they discontinued the free service.
- Alliance support ensured that two separate Boost releases were not late.

- Alliance staff has assisted with wowbagger administration since 2022.
- Alliance's IT department assists authors with support for continuous integration and build issues.
- Alliance's IT department assists release managers to troubleshoot and streamline the release process.
- Alliance provides the paid GitHub Actions self-hosted runners which provide build service during peak usage times.
- Alliance acted immediately and professionally to secure the boost.org domain upon learning it was at risk of loss.

Library and Project Support

- Creator and sponsor of the full-text search feature on boost.org.
- Sponsor of the [Boost.Unordered Development Plan](#).
- Sponsor of [Boost.Charconv](#) development and maintenance.
- Sponsor of [Boost.MySQL](#) development and maintenance.
- Sponsor of [Boost.Decimal](#) development.
- Sponsor of [Boost.Cobalt](#) development.
- Provides a full-time Senior Technical Writer for Boost.
- Created and maintains the [User's Guide](#).
- Created and maintains the [Contributor's Guide](#)
- Funds and operates the [Official C++ Slack Workspace](#), a home to many Boost library and project channels.

Strategic Vision

- Alliance President Vinnie Falco communicated his vision for improving the project in a blog post called [The Future of Boost](#).
- Commissioned a new, professional logo for Boost.

- Developed a modern website, customized for Boost, to create visual appeal and renewed interest for visitors.
- Promoted Boost at CppCon 2023 with brochures and posters.
- Promoting Boost at CppCon 2024 with new brochures and shirts.
- Posted all original messages to the Boost X account for the last 18 months.
- Alliance retains a digital illustrator to create unique artwork for each release, each newly accepted library, and [various places](#) on the new website, with the goal to establish a recognizable visual identity for Boost.

Fiscal Sponsorship Proposal

To continue carrying out the Alliance vision to revitalize Boost, a change in governance is needed. Our proposal is straightforward:

- The C++ Alliance becomes the fiscal sponsor for Boost under the terms of the Fiscal Sponsorship Agreement provided in Appendix 1.
- The C++ Alliance receives eligible donations for Boost, or donates directly to Boost, and custodies ownership of Boost's resources which currently include domain names, trademarks, and a server.
- The Steering Committee determines how resources are used.

This developers-first approach leverages the useful sponsorship model already developed by the Software Freedom Conservancy, and allows the C++ Alliance to minimize the risks incurred when making large investments in the project.

Transition Plan

A successful transition requires cooperation between the Boost Foundation, C++ Alliance, and the Current Committee (designated individuals who initially represent the project). Upon acceptance:

- Alliance launches the Beman Scholarship, if permission is given.
- The Boost Foundation is reimbursed for any and all reasonable legal expenses and estate fees incurred since the beginning of 2024.

- The Alliance assumes all costs and responsibilities for Boost's infrastructure, including but not limited to the wowbagger server, the mailgun service, and the new website.
- The mailing list will be migrated to the latest version, but the user experience of subscribers will appear unchanged.
- Reports apprising the community of the ongoing expenses, condition, and changes to these resources shall be published to the mailing list no fewer than three times per year.

The annual costs for the current website and the new website are estimated at \$13,000 and \$17,000 respectively. Even a conservative annual cost of \$20,000 a year represents less than 1% of the total expenses incurred by the C++ Alliance in 2023. While the new website was expensive to develop, the costs associated with its cloud services are not. Financial independence is a realistic goal for Boost.

Upon completion of this plan, Boost will be governed by stakeholders and have sufficient funding to maintain its infrastructure, ensuring its lasting success. Larger, more expensive projects will still come from volunteers as usual.

Summary

The aging and shrinking volunteer base of the Boost C++ Libraries poses headwinds to its long-term viability, as further declines in participation and momentum are likely without intervention. To address this challenge, it is essential that the project attracts new talent to restore energy and fresh ideas, yet the project's current governance stymies efforts to do so.

The C++ Alliance offers the vision, motivation, and resources necessary to revitalize Boost. We respectfully propose that you accept the Alliance as the new fiscal sponsor, and join us in safeguarding the longevity of the Boost C++ Libraries. Together, we can build a stronger, more sustainable foundation for the project's continued success.

Frequently Asked Questions

What problems is this organizational change attempting to solve?

The proposed Fiscal Sponsorship Agreement between the C++ Alliance and Steering Committee restores the power of the developers to make decisions for the project, and safeguards the Alliance's investments by ensuring that the use of Boost's assets are aligned with the best interests of the project.

What will the impact be for current volunteers?

Current volunteers will experience a beautiful new website, an upgrade to the mailing list, and new infrastructure which solves the risks posed by wowbagger's current end-of-life issues, as well as the benefits of Alliance's continued contributions to the project.

How does this proposal meet the changing needs of the Boost project?

This proposal brings with it the expertise and resources to help Boost transition its aging base of volunteers to new generations of younger volunteers who are interested in C++ and technical excellence.

How will decision makers be put in place?

Section §6 of the proposed FSA defines the process for establishing the Steering Committee in the same fashion as the previous agreement with Software Freedom Conservancy.

How will decisions be made?

Section §6 of the proposed FSA determines that the Steering Committee shall manage the technical, artistic and philanthropic direction of the Project.

What are the structure and bylaws of the Alliance?

The Alliance bylaws may be viewed at <https://cppalliance.org/pdf/Bylaws.pdf>.

The board of directors approves projects and provides oversight to ensure non-profit compliance and mission alignment, while the executive team works full time to carry out the mission.

Is the Alliance stable as an organization?

The Alliance has thrived with a small, stable board for years. To ensure continuity, the proposed FSA includes provisions allowing the Steering Committee to transfer the project to a new sponsor if necessary. The longevity of the Steering Committee is the more relevant metric of stability.

Is the Alliance taking over the libraries?

No one is taking over the libraries. The Alliance will serve in the same capacity as the Software Freedom Conservancy, with the same responsibilities and rights.

Is the Alliance taking over Boost's logo?

No one is taking over Boost's logo. The Alliance proposes a new logo to attract younger volunteers, and this becomes a donation controlled by the Steering Committee.

How will expenses be financed?

Expenses may be financed using one or more of the following:

1. Donations from the Boost Foundation.
2. Donations from individual supporters.
3. Donations from the C++ Alliance.
4. Sales of Boost-branded products such as apparel.
5. An endowment with Boost as the beneficiary.

Does the Boost Foundation have a stewardship responsibility to Boost?

The Foundation has no legal or contractual obligations to Boost. The assets owned by the Foundation such as the boost.org domain name, wowbagger server, and trademarks are the sole property of Foundation and may be used for any tax-exempt purpose.

Appendix 1

Fiscal Sponsorship Agreement

This area is intentionally left blank.

FISCAL SPONSORSHIP AGREEMENT

This Agreement is made by and between The C++ Alliance ("Alliance") and Ion Gaztañaga Muñoa, Joaquín M López Muñoz, and René Ferdinand Rivera Morell (the "Current Committee") on behalf of the project known as Boost (the "Project"). Alliance is a California-registered corporation, which has received recognition of exemption from federal income tax under Section 501(c)(3) of the Internal Revenue Code (IRC).

WHEREAS:

- Alliance's organizational mission and charitable goal includes making the C++ programming language accessible and useful to anyone who wishes to learn and apply the language, by supporting the development of high quality, expert reviewed, and legally unencumbered open-source libraries.
- The purpose of the Project is to produce, distribute, document, and improve software and/or documentation that can be freely copied, modified and redistributed, and for which modified versions can also be redistributed ("Free Software"), and to facilitate and organize its production, improvement and ease of use.
- Alliance desires to act as the fiscal sponsor of the Project beginning on the Effective Date (as defined below) to assist the Project in accomplishing its purpose, which Alliance has determined will further Alliance's charitable goals. The Current Committee desires to manage the Project under the sponsorship of Alliance.
- Alliance's Board of Directors has approved the establishment of a fund to receive donations of cash and other property earmarked for support of the Project and to make disbursements in furtherance of the Project's mission (the "Project Fund").

Currently, the principal office of the Project is located at: _____

Now, THEREFORE, THE PARTIES HEREBY AGREE AS FOLLOWS:

1. **Term of Agreement.** As of the Effective Date, the Project joins Alliance; this relationship will continue unless and until terminated as set forth in §8.
2. **Project Management and Activities.**
 - a. **The Steering Committee Will Manage the Project.** Authority to manage the technical, artistic and philanthropic direction of the Project and the program activities of the Project is delegated to the Steering Committee as defined in §6, subject at all times to the direction and control of Alliance's Board of Directors. Alliance will only intervene in the program activities to the extent the Project is not in compliance with §2(b) or §5 of this Agreement.
 - b. **The Project Will Be Free Software.** Alliance and the Steering Committee agree that any and all software and/or documentation distributed by the Project will be distributed solely as Free Software. Alliance retains the sole right to determine whether Project's software and/or documentation constitutes Free Software (as defined herein).
 - c. **Ultimate Responsibility of Project.** Subject to §2(a) of this Agreement, all community programs, public information work, fundraising events, processing and acknowledgment of cash and non-cash revenue items, accounts payable and receivable, negotiation of leases and contracts, disbursement of Project funds (including grants), and other activities planned by the Project shall be the ultimate responsibility of Alliance and shall be conducted in the name of Alliance, beginning on the Effective Date.
 - d. **Project Not An Agent Of Alliance.** The Current Committee hereby acknowledges that the Project and the Steering Committee do not and shall not act as an agent for Alliance unless specifically authorized in writing by Alliance to do so.
3. **Fees.** The Alliance agrees to bear all costs for general operations. Notwithstanding the above, the Project agrees that should Alliance be required to pay any taxes (including but not limited to sales taxes and unrelated business taxable income) as the result of any activity of the Project

and/or activities undertaken by Alliance on the Project's behalf, such taxes shall be deducted from the Project Fund. Alliance will monitor any unrelated business taxable income and may require the Project to cease activities generating such income if the overall amounts exceed amounts permissible or prudent for Alliance, given Alliance's tax exempt status.

4. **Project Fund/Variance Power.** Beginning on the Effective Date, Alliance shall place all gifts, grants, contributions and other revenues received by Alliance and identified with the Project into a Project Fund to be used for the sole benefit of the Project's mission as that mission may be defined by the Steering Committee from time to time with Alliance approval. Alliance retains the unilateral right to spend such funds so as to accomplish the purposes of the Project as nearly as possible within Alliance's sole judgment. Alliance agrees to make a good faith effort to consider any expressed donor intent in making determinations on the expenditure of that donor's gift; however, the Parties acknowledge that expressions of donor intent are not legally binding on Alliance. The parties agree that all money, and the fair market value of all property, deposited in the Project Fund be reported as the income of Alliance, for both tax purposes and for purposes of Alliance's financial statements. It is the intent of the parties that this Agreement be interpreted to provide Alliance with variance powers necessary to enable Alliance to treat the Project Fund as Alliance's asset in accordance with Financial Accounting Statement No. 136 issued by the Financial Accounting Standards Board, while this Agreement is in effect.
5. **Project Fund Management / Performance of Charitable Purposes.** All of the assets received by Alliance under the terms of this Agreement shall be devoted to the purposes of the Project, within the tax-exempt purposes of Alliance. The Project agrees not to use its funds or operate in any way which would jeopardize the tax-exempt status of Alliance. No item of revenue shall be earmarked for use in any attempt to influence legislation within the meaning of IRC Section 501(c)(3) and no agreement, oral or written, to that effect shall be made between Alliance and any revenue source. Alliance shall not use any portion of the assets to participate or intervene in any political campaign on behalf or in opposition to any candidate for public office, to induce or encourage violations of law or public policy, to cause any private

inurement or improper private benefit to occur, nor to take any other action inconsistent with IRC Section 501(c)(3).

6. **Representation of the Project in Alliance.** The Current Committee, each a signatory hereto, hereby establishes and comprises the initial members of the Steering Committee to represent the Project in its official communication with Alliance. The Current Committee hereby acknowledges that the Steering Committee will be subject to all terms of this Agreement. On the Effective Date, the Current Committee hereby transfers all rights, obligations and privileges of this Agreement over to the Steering Committee.

Existing Steering Members ("Members") can be removed from and new Members can be added to the Steering Committee by simple majority vote of the existing Steering Committee; however, three (3) shall be the mandatory minimum number of Members. All decisions of the Steering Committee shall be made by simple majority. The Steering Committee shall appoint, by majority vote, one Member as its "Representative" to communicate all Project decisions to Alliance.

The Representative shall promptly inform Alliance of changes in the Steering Committee composition and of contact information for all Members. If Alliance is unable, after all reasonable efforts, to contact a majority of the Members for a period of sixty (60) days, or if the minimum number of Members is fewer than three for a period of at least sixty days, Alliance may unilaterally appoint new Members from the Project community to replace any unreachable Members and/or to increase the Steering Committee composition to three Members.

Modifications to this section of the agreement must be approved by a three-fourths vote of the Steering Committee, and then proposed in writing to Alliance. Such modifications do not take effect until Alliance gives its written approval, such approval not to be unreasonably withheld.

7. **Outstanding Liabilities.** The Current Committee represents that any liabilities that may be outstanding in connection with the Project have been disclosed to Alliance.

8. **Termination.** The Steering Committee or Alliance may terminate this Agreement at any time subject to the following understandings:
- a. **Notice and Successor Search.** Either Alliance or the Steering Committee may terminate this Agreement on sixty (60) days' written notice to the other Party, so long as a Successor can be found that meets the following requirements (the "Successor has Qualified"):
 - i. the Successor is another nonprofit corporation which is tax-exempt under IRC Section 501(c) (3),
 - ii. the Successor is willing and able to sponsor the Project, and,
 - iii. the Successor has (a) communicated its willingness to sponsor the Project in writing to Alliance and (b) sent a copy of its 501(c)(3) determination letter to Alliance, and,
 - iv. the Successor is approved in writing by both Parties by the end of the 60-day period, such approval not to be unreasonably withheld.
 - b. **Additional Search Periods.** If the parties cannot agree on a Successor to sponsor the Project, the Steering Committee shall have an additional 60 days to find a Successor willing and able to sponsor the Project. Any subsequent search periods of any length shall only be granted at Alliance's written permission.
 - c. **Transfer to a Successor.** If a Successor has Qualified, the balance of assets in the Project Fund, together with any other assets held or liabilities incurred by Alliance in connection with the Project, shall be transferred to the Successor within thirty (30) days of the approval of the Successor in writing by both Parties or any extension thereof, subject to the approval of any third parties that may be required.
 - d. **Termination Without a Successor.** If no Successor is found, Alliance may dispose of the Project assets and liabilities in any manner consistent with applicable tax and charitable trust laws.

- e. **Current Committee's Right to Terminate.** The Current Committee hereby acknowledge that they will relinquish any rights to terminate separate from the Steering Committee as of the Effective Date.
- 9. **Miscellaneous.** Each provision of this Agreement shall be separately enforceable, and the invalidity of one provision shall not affect the validity or enforceability of any other provision. This Agreement shall be interpreted and construed in accordance with the laws of the State of California. This Agreement constitutes the only agreement, and supersedes all prior agreements and understandings, both written and oral, among the parties with respect to the subject matter hereof.
- 10. **Amendments.** This Agreement may not be amended or modified, except in writing and signed by both Alliance and the entirety of the Steering Committee.
- 11. **Counterparts / Facsimile.** This Agreement may be executed in two or more counterparts, each of which shall constitute an original, but all of which, when together, shall constitute but one and the same instrument, and shall become effective when one or more counterparts have been signed by each Party hereto and delivered to the other Party. In lieu of the original, a facsimile transmission or copy of the original shall be as effective and enforceable as the original.

IN WITNESS WHEREOF, the parties have executed this Fiscal Sponsorship Agreement effective on the ___ day of _____, _____ (the "Effective Date").

By: _____

Date: _____.

The C++ Alliance
Vincent Falco
Title: Executive Director

By: _____

Date: _____.

Ion Gaztañaga Muñoa

By: _____

Date: _____.

Joaquín M López Muñoz

By: _____

Date: _____.

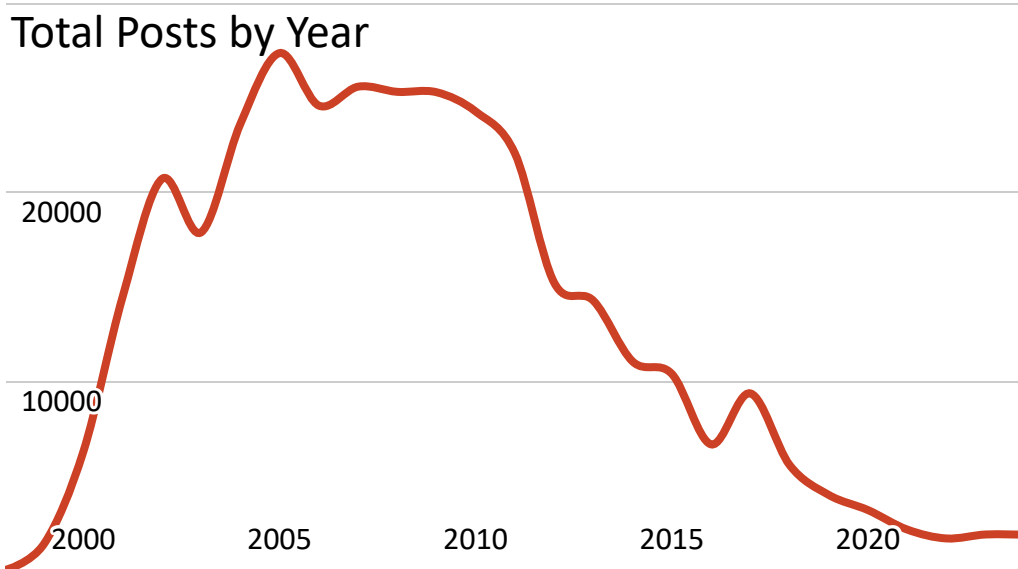
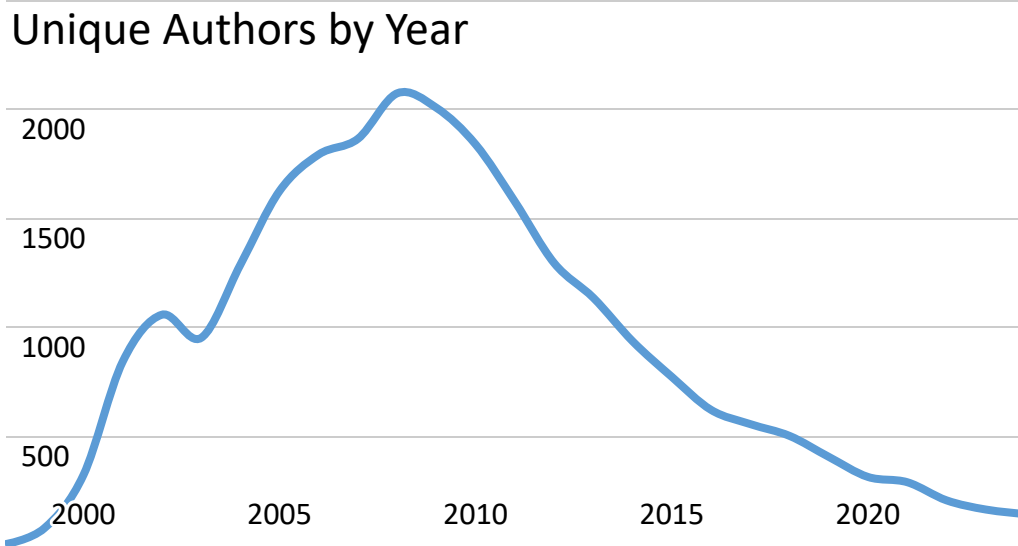
René Ferdinand Rivera Morell

Appendix 2

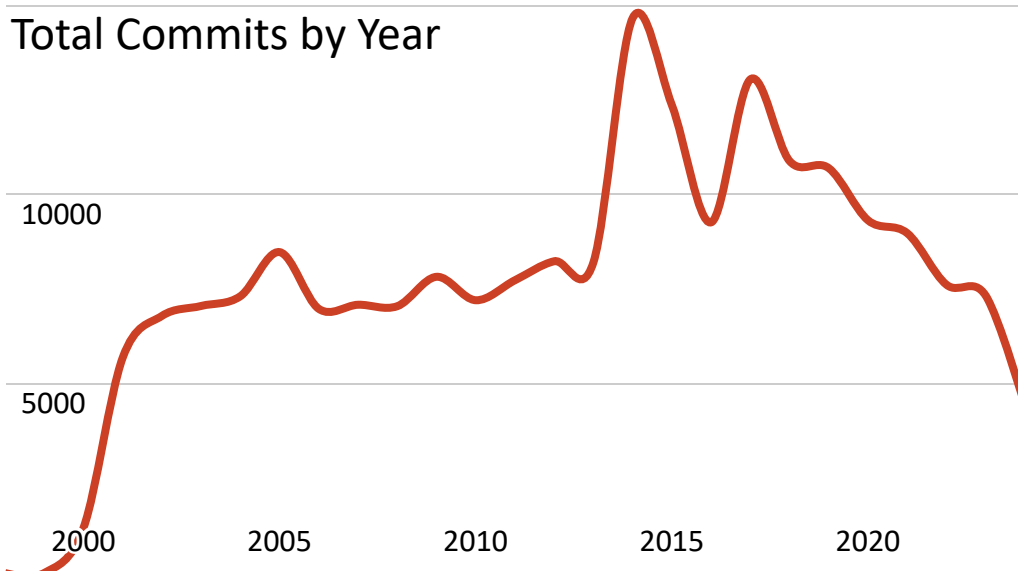
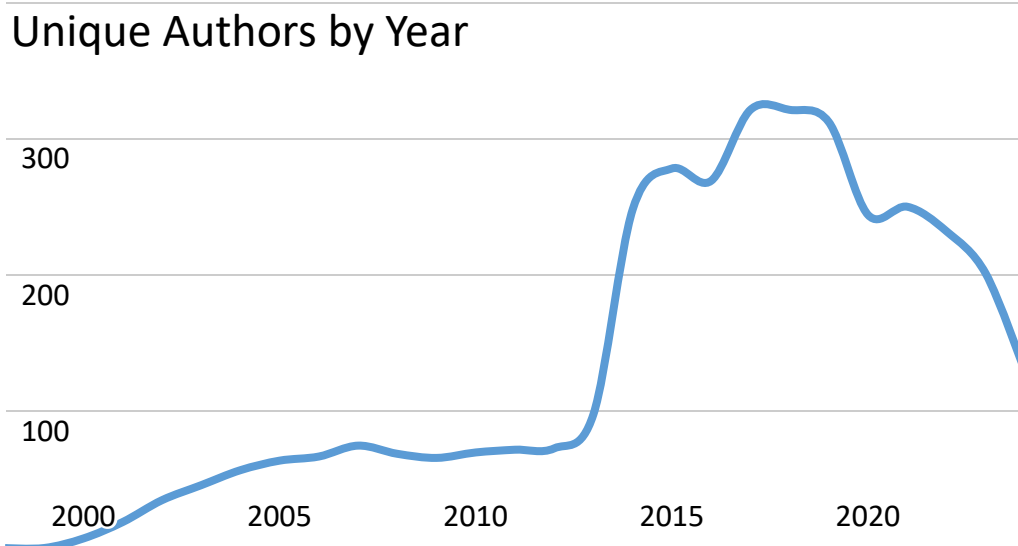
Boost Participation Key Metrics

This area is intentionally left blank.

Boost Mailing List Activity



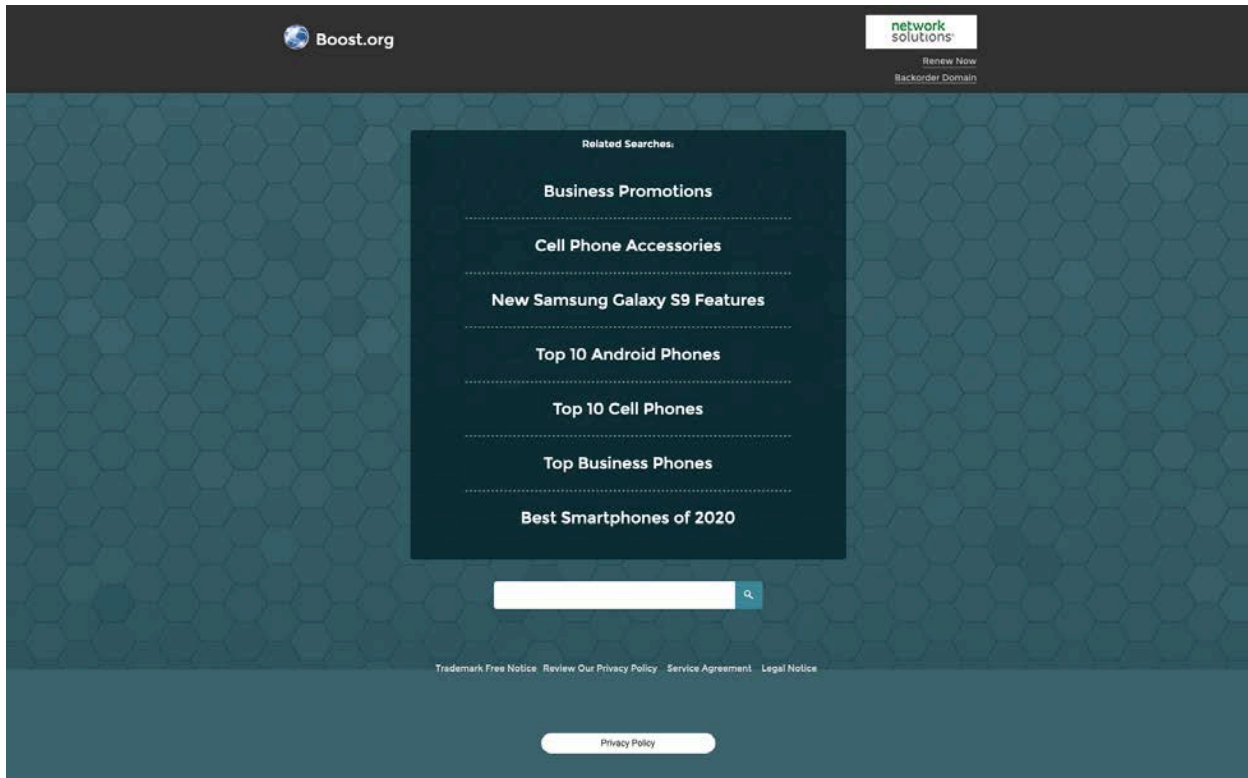
Boost GitHub Repository Activity



Posts	Authors	Year	Commits	Authors
46	8	1998	0	0
1447	81	1999	0	0
6323	325	2000	1189	7
14493	844	2001	5686	19
20707	1059	2002	6789	35
17889	953	2003	7052	46
23618	1288	2004	7310	57
27355	1626	2005	8474	64
24581	1793	2006	6975	67
25580	1865	2007	7084	75
25307	2073	2008	7046	69
25287	2006	2009	7822	66
24264	1838	2010	7203	70
22047	1576	2011	7732	72
15223	1295	2012	8233	73
14272	1136	2013	8207	98
11062	937	2014	14657	248
10406	774	2015	12353	278
6688	623	2016	9261	269
9380	558	2017	13037	321
5597	504	2018	10879	321
4015	408	2019	10690	312
3215	316	2020	9313	244
2187	292	2021	8972	250
1719	209	2022	7612	232
1929	167	2023	7322	201
1914	146	2024	4411	129

Appendix 3

boost.org Domain Registration Expiration



Appendix 4

wowbagger Server Report

This area is intentionally left blank



Vinnie Falco <vinnie@cppalliance.org>

Wowbagger

Sam Darwin <sam@cppalliance.org>
To: Vinnie Falco <vinnie@cppalliance.org>
Cc: Louis Tatta <louis@cppalliance.org>

Sat, Jul 6, 2024 at 5:07 PM

Wowbagger

The current boost.org web server is a CentOS Linux 6.10 cloud instance hosted at Rackspace. In general, the software packages and operating system are out-of-date and so it's important to consider the upgrade path.

To review specific details:

Costs: around \$200/month total, including 3 instances: ftp server, monitoring server, wowbagger webserver.

Web Server OS: CentOS 6 Release Date July 10, 2011 End of Life November 30, 2020

The operating system has been unsupported for nearly 4 years. No security fixes. There are progressively more difficulties with SSL certificates, because in order to process certs the latest CA (certificate authority) certs need to be available locally and they are not. The version of SSH is quite old. This can prevent logins to/from newer machines. It would not be easily possible to re-install the same server since CentOS 6 depends on remote yum repositories that are offline in 2024.

Website boost.org:

The website is based on PHP5, an outdated version of PHP that can't be installed on newer servers. Therefore, a direct migration to move the website onto Ubuntu 24.04 or similar is blocked. The website needs to be revised using PHP8. Since the codebase is a complex object-oriented system with dependencies, that requires careful development and testing. It's not impossible to upgrade to PHP8, but would take many weeks, may be error prone, and the appearance of the site would still be the same.

Mailman2 mailing list:

Mailman2 is no longer supported on the latest releases of RedHat and Debian/Ubuntu. Meaning, the Boost mailing list cannot be simply migrated or reinstalled on a newer server. Rather, the replacement is mailman3, a complete rewrite of the software using Django. Mailman3 is comprised of multiple components: mailman-core, Postorius, and Hyperkitty.

Rackspace: In the last few months, wowbagger has experienced two disk outages caused by a faulty NIC card (connecting to Network attached storage). Although recovery was successful each time, there is cause for concern since this server is not redundant (web1 + web2). If it crashes the boost website would go offline.

As can be seen, the current system is obsolete to the extent that it is not possible to simply upgrade a few packages and/or move the current services to another server. Everything requires a significant upgrade. That is what the C++ Alliance has been moving towards, with the work on boost.io, and testing of mailman3.

On Sat, Jul 6, 2024 at 3:07 PM Vinnie Falco <vinnie@cppalliance.org> wrote:

Sam:

Can you please tell me exactly what is going on with wowbagger? Does it have a failing disk? Can it be cloned or otherwise replicated? Why can't it be upgraded or split up into multiple services? How much does it cost to run?

Thanks



Mailing List

Sam Darwin <sam@cppalliance.org>
To: Vinnie Falco <vinnie@cppalliance.org>
Cc: Louis Tatta <louis@cppalliance.org>

Mon, Aug 12, 2024 at 4:04 PM

Current state of the mailing list:

Mailman2 is running on wowbagger, a CentOS release 6.10 server, along with hypermail archives. The system is generally functioning acceptably well, however it cannot be upgraded because when attempting to install mailman2 on the latest RedHat or Ubuntu, necessary dependencies and packages are not available anymore. Mailman2 development has been discontinued. The official upgrade path is mailman3.

What is involved in the proposed upgrade path to mailman3?

There are multiple steps to install mailman3.

- Launch cloud server instances
- Set up mailgun gateway, api keys
- Install mailman3. In-house Ansible scripts will deal with much of the complexity of installing the database, nginx web server, elasticsearch backend, directory structure, etc.
- Create lists in mailman3. "Boost". "Boost-announce". "Boost-users".
- Import the lists "/opt/mailman3/bin/mailman import21 ..."
- Configure the lists, in settings.
- Import the message archives "mailman-web hyperkitty_import --since 1970..."

Why do we need mailman3?

See "current state of the mailing list" above. The wowbagger server is end-of-life.

How much different is it from the previous version?

Mailman3 is a complete rewrite, from scratch. Everything is new and different. However the user experience of using a mailing list should remain mostly familiar. Mailman3 has three main components:

Hyperkitty - a web portal, with a forum-like interface, hosting the archives.

Postorius - also running on Django webserver, as Hyperkitty does, this package allows subscribing and managing your membership on the lists.

Mailman-Core - the backend. Sending and receiving email messages.

Can we stay on the previous version?

No. See "current state of the mailing list" above.

Why isn't this just a one-click update?

It is not, see the section "What is involved in the proposed upgrade path to mailman3?"

On Mon, Aug 12, 2024 at 4:27 PM Vinnie Falco <vinnie@cppalliance.org> wrote:

Sam:

Can you please email me back with a summary explaining the current status of the mailing list, and what is involved in the proposed upgrade path to mailman3? Why do we need mailman3? What does it represent, how much different is it from the previous version? Can we stay on the previous version? Why isn't this just a one-click update?

Thanks

Appendix 5

The Future of Boost Blog Post

This area is intentionally left blank.

The Future of Boost

By [Vinnie Falco](#) on May 8, 2023

Greetings! I'm Vinnie Falco, Boost library author, C++ enthusiast, and the founder of The C++ Alliance, a 501(c)(3) non-profit. While some of you are enjoying the C++Now conference this week, I'd like to share some background on our organization and some history, outline a vision and goals for C++ and Boost, and solicit your feedback and support.

How It Started

I took notice of the C++ Standards Committee ("WG21") while I was writing Boost.Beast in 2016. Howard Hinnant, a co-workers at Ripple, taught me about writing papers and committee meetings. Beast used Boost.Asio (portable networking for C++) and I was and still am a huge fan of this network library. I learned that Asio was being proposed for standardization. There was even a "Networking TS" document: Asio was very close to becoming part of C++ officially! But the author Christopher Kohlhoff always seemed to not have the time to attend the meetings and push this proposal though.

Something which should not surprise anyone is that I despise paying taxes. In 2017, I had an idea: create a charitable organization which I can donate pre-tax income to, and then I could hire Christopher Kohlhoff as a "staff engineer" to work full time on C++ standardization, and Boost things! I would find the very best C++ people who are already doing open source work, then hire them full-time so they could focus on their open source C++ open work from home, instead of traveling to a boring job in order to make a living.

A Few Setbacks

In 2018 I offered this opportunity to Chris and he surprisingly turned it down. He actually liked going into an office and interacting with customers and users. He explained that the evolution of Asio and his WG21 work is not bottlenecked by time. Instead, he prefers to "think deeply

about things over a long period, and then write something.” Basically the opposite of my strategy, which is to write a bunch of code quickly and then throw out the bad parts.

This is a setback but I am not so easily deterred so I offered the same opportunity to Peter Dimov, an engineer of immense talent whose libraries are legendary. He also declined, explaining that taking a salary would transform a hobby into an obligation, affecting the quality and enjoyment of the work.

Now I’m thinking, well this is a disaster! We had the non-profit in operation officially since March of 2018 (the IRS approved us in September of 2019). We had the C++ language Slack workspace as of November of 2017, transitioned to a paid plan with full history. Our strategy shifted to focus on supporting the Boost Library Collection directly. We hired our first Staff Engineer, Marshall Clow, in April of 2018.

Fast forward and today we have 11 staff members. We have a great CTO/DevOps genius Sam Darwin. And we have Louis Tatta, our CEO that keeps things running smoothly and helps get the most out of every dollar donated. At some point I’ll share a complete list of everything that The C++ Alliance has done since the beginning, but that is the subject of another missive. Today I would like to talk about a vision for Boost.

The Boost Library Collection

Long-timers know Boost’s history but for those that don’t, Beman Dawes and Robert Klarer came up with the idea of a website offering curated, high quality C++ libraries in May of 1998. They described the “Formal Review,” a social technology where a group of peers would go over a proposed library at an agreed-upon time. They could interrogate the author about the library on the mailing list, and debate things. The outcome is a collection of posts where each reviewer summarizes their critique of the library, including whether or not to “accept” the library (sometimes with conditions). The founding documenting evokes a feeling of something big:

<https://www.boost.org/users/proposal.pdf>

The collection was named “Boost” and received many great contributions. The authors and reviewers were active in the standardization committee. In December of 2005, Boost.Asio was added after being developed since 2003. In September of 2011 the C++11 standard was published, containing many library components modeled closely or identically to their Boost counterparts. In my opinion, Asio’s efforts at standardization were thwarted by the growth of politics; an inevitable consequence of the bureaucratic ISO structure.

Boost launched its own conference called BoostCon in 2007 on the heels of its success. Speakers included Scott Meyers, Dave Abrahams, Eric Niebler, Howard Hinnant, and other

juggernauts of C++. A new conference called CppCon was launched in 2014 and attracted even larger crowds, as it was focused on C++ in general.

Trouble Brewing

With the release of C++11, there were now components in Boost which were duplicated in the Standard Library. The C++ committee became more popular and valuable owing to the success of C++11, made possible in part by years of lead-up from the talented Boost engineers. The conferences turned some people into the equivalent of pop stars, appearing as staple keynote speakers.

Library writers discovered it was easier to get a proposal into the C++ standard than it was to get a library through the Formal Review process. They discovered that there was more glory to have a proposal accepted into the official C++ language, than to have their library accepted into Boost. And once their proposal became part of C++, they no longer had to “maintain their code” the way they would if their library got in Boost. A Formal Review evaluates the author somewhat in addition to the library. Because once a library is accepted, it must be maintained and evolved. When an author abandons their Boost library, the community is impoverished as the knowledge and expertise leaves with them. And someone else must take over the maintenance.

In December of 2020, Beman Dawes passed away and Boost suffered a loss which can never be replaced. Beman had an enormous impact on not just the libraries but also C++. He was in WG21 from the very beginning, chaired LWG (the “Library Working Group”) for quite some time, and achieved a long history of open source contributions.

Boost had other problems. Fewer libraries were being proposed, and it took longer to find a volunteer review manager. Mailing list volume declined steadily. At 160+ libraries, users complained that “Boost is too large.” They complained that “many of the libraries are outdated”, that “the documentation is of varying quality”, and that “Boost takes too long to compile.” They complained about the obscure build system and lack of cmake support. The archetype of “never Booster” appeared: individuals or corporations who ban the use of Boost entirely.

Beman was the closest thing resembling a leader, despite Boost being a federation of authors with each having final word over their own library. Beman would solve problems, help with the direction of things, and even “beat the bushes” when it was time for a review by reaching out to his network of contacts and soliciting their participation. Without Beman, Boost lost its leader. Boost lost its Great Founder. And no one has since filled the role.

The C++ Alliance

At this point, a vision for what our non-profit could do crystallized. We would help C++ in general by refreshing the foundations of Boost, restoring Boost's prominence in the community, and helping Boost become a leader once again as innovators in C++. To do this, I'll share what we feel are the problems facing Boost, and ways to address some of them. Finally I'd like you to weigh in on all of this and help figure out what is important and what successful execution might look like.

We believe Boost faces these obstacles, organized broadly by category:

Stagnation

- There are fewer new libraries proposed.
- Formal reviews get less participation.
- Review managers are typically scarce now.
- The mailing list volume is thinning; younger folks don't use lists.
- There is no second order effect: new libraries rarely use Boost.

Quality

- Some libraries are unmaintained and create a negative user experience.
- Users open issues, and no one replies to them.
- Pull requests are submitted to abandoned repositories.
- Scant financial resources for infrastructure or staff.

Documentation

- The quality of documentation varies greatly across libraries.
- The rendered pages and content of some documentation looks dated.
- Some toolchains used are obscure and unmaintained.

Perception

- Boost causes long compile times.
- The libraries have too many interdependencies
- Supporting old C++ versions is a weakness not a strength.
- The duplication of std components is wasteful and causes friction.
- The "Monolithic" distribution of Boost is obsolete.

Messaging

- The website is outdated and never receives updates.
- Boost's value proposition is not clear ("why use boost?")
- There is no clear voice countering misconceptions and irrational phobias.

- Users receive no guidance about the future, or what is maintained.
- The libraries have no representation at conferences.

Users have also weighed in with their thoughts on Boost:

https://www.reddit.com/r/cpp/comments/gfowpq/why_you_dont_use_boost/

A Plan

I love C++, supporting users, and the Boost author experience. I think these problems can be solved. But not by demanding that “everyone who maintains a Boost library must do X.” In Boost culture when you want something done you need to do it yourself, then convince the mailing list of the merits of your proposal.

As a library author and contributor, I know that whatever I do will never rise to the same level as the original act of the creation of the Boost Library Collection. But I will be satisfied if I can stoke its fires and bring them back to a roar. To this end the resources of the non-profit are directed into projects we believe will positively affect Boost:

Website Renovation

Our vision for an updated Boost website is clean and stylish, which speaks to a large and diverse audience. This site will have a design and content that effectively communicates the value proposition of using the Boost Library Collection: that you will write better C++ code, become more productive, and achieve greater success in your career or personal projects. Features will foster participation and revisits, with content updated regularly. The library presentation is elevated with a new visual design language that evokes distinction and appeal, and credits the authors, maintainers, and contributors that bring it to life.

To achieve this vision, you have probably heard that we contracted an outside software firm to build something precisely tailored for our needs. We care too much about Boost to use an ill-fitted, off the shelf product. This website has a lot of software behind it (written in Python as part of a Django framework application) and like most software projects it is late and over budget. I’ll refrain from saying “it’ll be ready soon” and just post a link to the new site instead, hopefully in a few weeks.

I have been personally involved in the design, presentation, and execution of the features of the website, most of which have been cut from the initial release in order to speed things along. The goal is to show the library collection in a way that highlights its strengths and speaks to a desire of every C++ programmer: to find the perfect library they can add as a dependency to help complete their next project.

The Boost website and the site documentation can be illustrated by retaining a talented digital artist to produce custom assets that are unified in style, colors, and messaging, so that the entire site feels purposeful. This artist will also provide imagery used for our social media campaigns such as the announcements we make on Twitter which some of you might have already seen

https://twitter.com/Boost_Libraries

I have strived to give every tweet an image to enhance the Boost brand.

Recently an animated discussion on the mailing list took place about adding a forum which does not replace the mailing list but is integrated to work with it. Posts in the forum become posts to the mailing list, and vice versa. Users of the mailing list and users of the forum will have no idea they are interacting, even though they are. This can only be possible if we write the software ourselves, from the ground up, with exactly one constraint: the mailing list will continue to operate exactly as it does today, on an unmodified version of Mailman 3. The mailing list users stay happy, and we can attract new people who prefer a web-based interface.

The C++ Alliance prioritizes its allocation of resources to ensure not only the website's completion, but also dedicated staff for ongoing maintenance and improvement. The Boost website will rise over time to the same level of quality expected of every Boost library. Community members should feel free to open issues on the website repository with bugs or features, knowing that every issue will be looked at, triaged, and addressed appropriately.

Documentation Improvement

Our vision for documentation is to ensure that every Boost library has the option to adopt a well-maintained toolchain that is easily deployed, produces high-quality output befitting the Boost brand, is itself well-documented and easy to use, and has behind it full-time staff working continuously to make improvements and provide technical support.

After researching the domain extensively (by just asking Peter Dimov) we have discovered that the markdown format AsciiDoc is a very popular format with a simple and well maintained toolchain. Several regularly active Boost authors have already switched their libraries to using AsciiDoctor. The authors of the AsciiDoctor tool are also the authors of "Antora," a modular, multi-repository documentation site generator:

<https://docs.antora.org/antora/latest/>

We have built a new, modern set of additional scripts capable of building the Boost release and documentation, including the capability of rendering "Antora-enabled Boost library repositories" using this Antora system. The results are beautiful and modern, and the

Asciidoctor / Antora toolchain holds the promise of being popular and well-maintained for a long time. The use of Asciidoc or Antora is optional; this is just an additional choice.

Peter Turcan is our full-time Senior Technical Writer who is modernizing the instructions for users, maintainers, contributors, and formal review participants. You can see Peter's work along with the quality of Antora's output here (note that the user-interface is stock and will be restyled soon):

<https://docs.cppalliance.org/>

The website above has a new full-text search feature (try it!). We are investing in a search experience which includes the site docs, library docs, library references, and even the public header files. We are also investing in the deployment of a large language model (ChatGPT-style AI) trained in Boost and C++ specifics to answer questions for users. We have a new talented and eager staff engineer working full-time exclusively on this, and I don't want to steal his thunder so I will let him explain further soon.

Some Boost libraries currently generate their documentation reference pages using Doxygen combined with other obscure tools such as xsltproc or Saxon-HE to render into Boost Quickbook, an obsolete form of markdown which only we use. This Quickbook is rendered into BoostBook, which is a flavor of DocBook. The BoostBook is converted into HTML by a DocBook renderer. This rapidly obsolescing toolchain is painful to work with and is a form of technical debt which costs us.

I have begun work on a new command-line tool called MrDox ("mister docs") which uses the unstable clang libtooling API to extract the documentation comments and declarations from C++ programs, and turn them into beautiful Asciidoc reference pages. You can see that work here:

<https://github.com/cppalliance/mrdox>

The core principles of the design of MrDox is to always understand the very latest C++ constructs and extract them with high fidelity. For example it recognizes conditional noexcept, constexpr, deduction guides, all attributes, and many other things that other documentation toolchains cannot fathom. In a nutshell I intend to bring the same level of Boost quality to the documentation toolchain that Boost has brought to the C++ libraries themselves.

MrDox intends to completely replace Doxygen, xsltproc, Saxon-HE, Quickbook, Boostbook, and Docbook, as the only requirement to render its results is to run the Asciidoctor tool, which has no other dependencies. This toolchain offers modernization and simplification for anyone who opts-in to it, which reduces long-term risks and improves results. This unfortunately delays the development of my other libraries, but enhancements in the documentation toolchain are a force multiplier; many Boost libraries can benefit.

Continuous Integration

Our vision for continuous integration is to bring the most talented individuals together and combine that with state of the art technology and resources to ensure that every library has at its disposal, access to robust cloud services for continuous integration. These services are the lifeblood of maintaining and communicating the quality of a library. We aim to provide dedicated staff and technical support to fortify Boost in the ever-shifting landscape of free CI services for open source projects.

The infrastructures providing our continuous integration services are the lifeblood of maintaining the high quality of the Boost collection. Library authors test against many versions of C++ and many different compiler versions. And we have many libraries; over 160 of them which all compete for the finite public resources offered by GitHub through GHA, through Azure Pipelines, or Appveyor.

When Travis discontinued its free service, our CTO Sam Darwin deployed Drone (<https://www.drone.io/>) instances and offered every Boost library a pull request which compiles and runs their tests on our new infrastructure. Although this service is still active and offered today, we are not content to leave it at that. CI services are volatile over time. Some come, some go, and some become overloaded which is the current situation with the public GitHub Actions runners during peak times. The Boost organization GitHub account has over one hundred and sixty libraries each submitting sometimes enormous numbers of jobs which take multiple hours to complete.

Although the GHA environment resources are subjected to recurring oversubscription, we feel that it offers the best framework for composable actions and flexibility. Sam is exploring the possibility of having self-hosted C++ Alliance runners dedicated only to Boost jobs during peak times. Ensuring high availability of CI resources is an ongoing project for us, and we are always evaluating existing and new solutions to provide the best-of-class choices for libraries.

Library Enhancements

Our vision for the libraries themselves is to preserve unchanged the amazing social technologies invented by the Boost founders which include the Formal Review process, the Release Schedule, the mailing list discussions, and the federated library ownership model. We want to ensure that no library is unmaintained and that every opened issue receives a response. We want the community to respect and admire the formal review process and participate with eagerness not only as reviewers but also as volunteer review managers and participants in the sometimes-heated list discussions. Library membership in the Boost library collection should be seen as the highest level of honor and recognition of effort.

The C++ Alliance has ongoing direct investments in improving existing Boost libraries and writing new ones to be submitted for formal review. Many folks are already aware of the optimization efforts being applied to the Boost.Unordered library, whose plan was written up by Peter Dimov. Joaquín M López Muñoz is providing his mathematical expertise and container experience, while Christian Mazakas (one of our full-time staff engineers) is writing the implementation, tests, and documentation according to specification.

People following Boost.Math might recognize Matt Borland as a regular contributor. He has joined us as a staff engineer and is currently working on a new library to be proposed for Boost: charconv, which is a C++11 port of the eponymous C++17 feature. This library will help libraries and users who may not have access to C++17 enjoy the same features through Boost instead.

Messaging and Direction

Our vision for Boost includes clear messaging to inform the public on the status of the libraries, the challenges we are facing, and what our future direction might be. We believe in robust two-way communication between library authors and maintainers, and the stakeholders which are largely the people and companies which use the Boost libraries. We believe in having a social media presence that helps convey the prestige and status that comes with the quality Boost libraries offer.

Currently we have only anecdotal evidence of Boost's adoption (or lack thereof) in various companies and projects. We only hear from the people who complain or open issues, or post to the mailing list. We do not have a concise list of companies using Boost, when new companies adopt Boost, or when companies stop using Boost. We do not have feedback from stakeholders about which Boost libraries they rely on the most, what they would like to see in future versions, or in some cases even if they are having problems with a library or its documentation.

The decentralized model of Boost library development works great for the problems it tries to solve but offers no overall directional guidance for Boost. Today the C++ language is facing unprecedented challenges: the popularity of Rust, the demands for "memory safety", the rise of Artificial Intelligence capable of writing software independently, and possibility that the bureaucratic structure of WG21 renders it incapable of meeting these challenges in a lively or effective manner.

We believe that Boost can offer the greatest value by focusing in the areas where C++ is strong and without meaningful competition. These include space exploration, game development, high-performance computing, embedded systems, the Internet of Things,

robotics and industrial process control, financial services, computer vision and graphics, scientific simulation, and more.

Furthermore the stunning and continued lack of networking in the standard library creates an opportunity for Boost to offer full-stack solutions in areas that speak to the strengths of C++. This is made possible because Boost already offers portable networking through Asio, HTTP and Websocket through Beast, excellent JSON parsing and serialization tailored for network programs, URLs, and more recently a Redis client (Boost.Aedis) and even a MySQL / MariaDB client. We intend to sponsor the development of non-Boost, open source applications and services that target specific underserved markets that would benefit from C++ solutions which use the excellent libraries that exist in Boost.

Where Do You Fit In?

Our vision, our activity, and our deployed solutions are all “opt-in.” No one controls Boost or its libraries. Change is only possible with consensus of the folks that matter: authors, maintainers, and release managers. If Robert Ramey wants to keep his documentation in hand-written HTML that is entirely his choice; no one dictates what libraries do. We can only offer choices, and hope they will be seen as valuable.

This has been a long read, and I appreciate your investment of time. How do you feel about this vision? What would you change, or add, and what needs work? We welcome feedback, and value the volunteers who share our vision and help us execute it.

I invite you to stay tuned for more great news, coming soon!

Respectfully Yours

Vinnie Falco

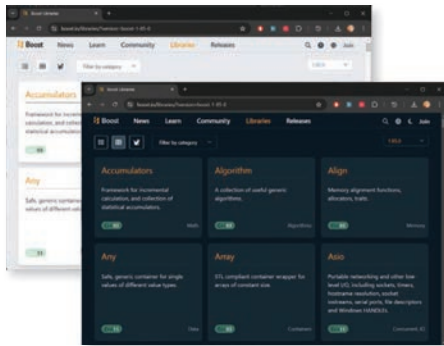
Appendix 6

Promotional CppCon Brochures and Posters

This area is intentionally left blank.

3.77 in w

Check out our new website at <https://boost.io>



Celebrate 26 years of Boost with us!

New Logo, Same Excellence

Our rigorous workflow enables hundreds of contributors working independently to create a single, well-tested release three times every year. Boost's founding principles deliver value for everyone using C++:

- Formal Review Process
- Boost Software License
- Robust Expert Discussion
- Broad Toolchain Support
- Technical Excellence
- Quality Control

back

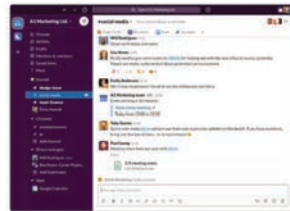
3.86 in w



The Official C++ Language Slack Workspace

<https://cpp.al/slack>

- Live chat for C++ community
- All countries and time zones
- Meet both novices and experts
- Home to Boost contributors
- Video and audio group calls
- Phone, browser, and desktop
- Full message and media history
- GitHub integration, and more.



Slack is a registered trademark and service mark of Slack Technologies, Inc.



The C++ Alliance is a financial sponsor of Boost and a Bronze supporter of CppCon 2024. Visit us at <https://cpp.al>



front

3.86 in w



The Boost C++ Libraries

<https://boost.org>

- Over 150 great libraries
- Three releases per year
- Often preinstalled
- Written by experts
- Fully documented
- Continuously tested

Using Boost will make your code portable, save you time, and increase your productivity at work or at home.

In the last seven years the collection has been revitalized by fresh volunteers and ideas. Join us as we discover the new Boost.



3.77 in w

New Website!

Boost's website has a brand new look which reimagines and improves the features of the previous site; including artwork lovingly drawn by a human, a database, GitHub integration showing insights into each library repository, and more, including:

- User Guide
- Contributor Guide
- Formal Review Guide
- Light and Dark Modes
- Library Documentation
- Track Release Schedule
- Full-text Search
- Live News Feed
- Option to Sign Up

Keep up to date by following us on X at https://x.com/@Boost_Libraries



Boost Has Something for Everyone

Networking

Write portable, optimized web applications using coroutines (since C++20), fibers, or callbacks, without waiting for the Standard Library to catch up.

- Boost.Asio** networking lets you connect to the Internet easily and efficiently
- Boost.Beast** speaks HTTP and WebSocket with an expertly designed interface
- Boost.Redis** is a high performing client with exceptional scaling
- Boost.MySQL** connects to external databases with an idiomatic API
- Boost.JSON** implements the web's standard for data interchange
- Boost.URL** does fast parsing and manipulation of URLs



Innovation

Boost delivers code, not papers; our libraries bring value to users by elegantly solving today's common problems.

- Boost.Describe** brings practical reflection since C++14
- Boost.Unordered** is a family of best-in-class hash containers
- Boost.Mp11** offers powerful metaprogramming tools since C++11 and later
- Boost.Cobalt** makes C++20 coroutines easy with algorithms needed to succeed
- Boost.Parser** will deliver modern, easy to use combinators (in 1.87.0)
- Boost.Scope** runs your code at scope exit. Based on P0052R10



Compatibility

Stuck on older toolchains? These all work since C++11:

- Boost.Assert** has `<source_location>`
- Boost.Charconv** provides `<charconv>`
- Boost.Variant2** is a never valueless `<variant>`
- Boost.Optional** is like `<optional>` and supports references!



Science

Boost is the industry leader for scientific applications where performance and correctness are paramount. These algorithms do the work so you don't have to.

- Boost.Histogram** is blazing fast and easy-to-use
- Boost.Math** speaks CUDA so good, even the STL uses it!
- Boost.Multiprecision** has big numbers for big problems
- Boost.Geometry** contains precision-engineered spatial algorithms



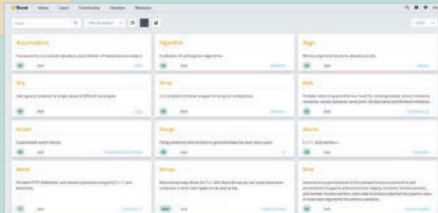


New website design

In progress at preview.boost.org



boost.org is getting its first revamp in 25 years! In addition to cooler looks, the new website will offer user log-in with GitHub or Google credentials, web-based forums tightly integrated with the classical mailing lists, updated tutorials, news section, and more!



Integrated full-text search

The embedded widget indexes documentation from all libraries to provide comprehensive, precise and structured search beyond what's possible with external engines.



Modularization

A recurrent request from users is the possibility to install individual Boost libraries to minimize HD space and build times. This is already provided by **vcpkg** and we're working with **Conan** to get there too.

Behind the scenes, a lot of work is underway to support modular builds by major package managers:

- Circular dependencies between Boost libs have been eliminated since 1.77 (Aug 2021).
- The **boostdep** tool automatically extracts the internal dependencies of any library.
- Lib repos are being updated so that **B2**, the official Boost builder, can work with them separately without requiring a full superproject download.

Stay tuned for further advances in modular installs!

CMake support

In development



Git clone Boost and use standard CMake procedures to build, install and use it. Integration in your builds is supported via **find_package**, **add_subdirectory** or **FetchContent**, either for the entire Boost superproject or for individually specified libraries.



Boost



The most widely used collection of high quality, peer-reviewed C++ libraries on the planet.

The libraries continue to thrive, delivering unparalleled performance and reliability.

[New website](#)

[Full-text search](#)

[Smaller installs](#)

[C++03 deprecation](#)

[Discussions groups at \[cplang.slack.com\]\(https://cplang.slack.com\)](#)

www.boost.org



The Boost mission is threefold:

- Develop high-quality, expert-reviewed, open-source C++ libraries
- Incubate C++ standard library enhancements
- Advance and disseminate C++ development best practices

Boost is guided by our shared values of transparency, inclusivity, consensus-building, federated authorship, and community-driven leadership.



Boost.Mp11



Fast, useful metaprogramming for the everyday programmer. Based on C++11 template aliases and variadic templates, **Boost.Mp11** makes it easy to automatically generate test cases for generic code, manipulate type lists as comfortably as run-time containers, or, in combination with **Boost.Describe**, visit classes for serialization or JSON conversion.

Boost.Describe



You don't have to wait for the next decade to enjoy C++ static reflection capabilities. Annotate your types with simple `BOOST_DESCRIBE_*` macros to automatically get a wealth of type information at compile time: member types and signatures, names, base classes... Best used in combination with **Boost.Mp11**.

```
#include <boost/describe.hpp>
#include <boost/mp11.hpp>
#include <cstdio>

enum E
{
    v1 = 11, v2, v3 = 5
};

BOOST_DESCRIBE_ENUM(E, v1, v2, v3)

int main()
{
    boost::mp11::mp_for_each<
        boost::describe::describe_enumerators<E>>([](auto D) {
        std::printf(
            "%s: %d\n", D.name, D.value );
        });
}
```

Boost.MySQL



Access your **MySQL** and **MariaDB** servers in a truly asynchronous fashion, using **Boost.Asio**'s async model.

Featuring a static interface that allows parsing rows into your own data structures, and a dynamic interface for maximum flexibility.

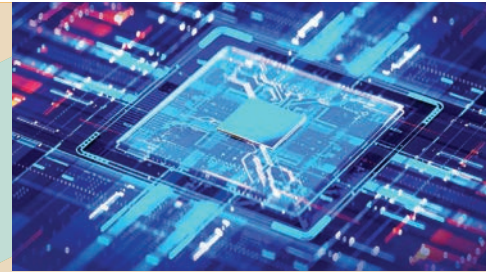
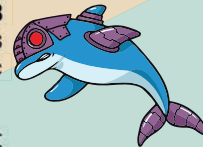
All of this with zero dependencies on the official C drivers!

Boost.Redis

Coming in **Boost 1.84 (Dec 2023)**

High-level client library for **Redis 6** or higher based on **Boost.Asio**.

Designed with a focus on performance, **Boost.Redis** allows for concurrent requests, pipelining and in-place data retrieval. STL containers and user-defined types can be easily managed through string serialization.



Boost.Unordered



In addition to standards-compliant implementations of `std::unordered_map` and relatives, **Boost.Unordered** now features:

boost::unordered_flat_map: One of the fastest hashmaps in the market. Takes advantage of open-addressing techniques and SIMD instructions to easily outperform standard containers by 3x or more.

boost::concurrent_flat_map: Suitable for high-load concurrent scenarios. To avoid deadlocking issues with iterators, this container introduces a novel, iterator-free API based on the visitation paradigm.

Boost.URL



Full implementation of **RFC3986** URI/URL specification.



Parse URLs into their constituent parts and use the container-based API to analyze them and synthesize new ones with ease.

Candidate Boost.Charconv

Upcoming review



Use `<charconv>` in your C++11 projects **today!** Convert from a sequence of characters to any integral/floating point type and back, with maximum speed and precision.

```
#include <boost/charconv.hpp>

const char* buffer = "42";
int v = 0;
auto r = boost::charconv::from_chars(
    buffer,
    buffer + std::strlen(buffer), v);
assert(r.ec == std::errc());
assert(v == 42);

char buffer[64];
int v = 123456;
auto r = boost::charconv::to_chars(
    buffer,
    buffer + sizeof(buffer) - 1, v);
assert(r.ec == std::errc());
// Strncmp returns 0 on match
assert(!strcmp(buffer, "123456", 6));
```

Boost



The most widely used collection of high quality, peer-reviewed C++ libraries on the planet

Input/output

JSON	JSON parsing, serialization, and DOM
Serialization	Data structure persistence and marshalling
ProgramOptions	Command-line argument parsing

Data structures

Unordered	Fast regular and concurrent hashmaps
Multindex	Containers with several view-like access interfaces
Intrusive	Intrusive containers and algorithms
Graph	Generic graph data structures and algorithms

Networking

Asio	Networking and other low-level I/O
Beast	HTTP, WebSocket, and network operations with Asio
URL	URL parsing and manipulation

System utilities

Interprocess	Process-shared resources and communication
MPI	Message passing for parallel applications
Process	Portable process creation and management

Metaprogramming

Describe	C++14 reflection through simple annotation macros
Mp11	Easy and fast C++11 type metaprogramming
Hana	Metaprogramming with heterogeneous sequences
PFR	Basic reflection for user-defined types

Math and Numerics

Math	Special functions, numerical algorithms and more
Geometry	Geometric algorithms, primitives and spatial index
Histogram	Fast multi-dimensional histogram

...and more!

150+ libraries and growing
BSL 1.0: the most permissive SW license in the market. No binary attribution required!
Available for self installation and on your system package managers, vcpkg and Conan
CMake support



We are working on a new website for Boost! Check it out at preview.boost.org



www.boost.org



Appendix 7

Fastly Invoice

This area is intentionally left blank.



P.O. Box 78266
San Francisco, CA 94107

INVOICE

Customer: The C Plus Plus Alliance, Inc.
5716 Corsa Ave Suite 110
Westlake Village, CA, 91362
US

Invoice Date: 8/1/2024
Invoice Number: 229935291
Account Number: 186kGEIrnzDFrQGLfp1HMMW
Usage Period: 7/1/2024 - 7/31/2024
Payment Terms: Due Upon Receipt - Credit Card
Purchase Order #:



Invoice Total:
\$ 2,770.00

Due Date: 8/1/2024

INVOICE SUMMARY

Item	Units	Rate	Amount
Global Bandwidth	71596.490311849 Gigabytes	\$ 0.0000000	\$ 0.00
Global Requests	130.4735 Units of 10,000	\$ 0.0000000	\$ 0.00
Full Site Delivery	1	\$ 2,770.000000	\$ 2,770.00
Fastly TLS - Non-Profit CA / Bring Your Own Certificates	3	\$ 0.0000000	\$ 0.00
Charges Total:			\$ 2,770.00
Sales Tax:			\$ 0.00
Invoice Total:			\$ 2,770.00
Previous Balance:			\$ 2,770.00
Previous Payments:			\$ -2,770.00
Account Balance:			\$ 2,770.00



Appendix 8

C++ Alliance Profit and Loss Statement

This area is intentionally left blank.

The C Plus Plus Alliance, Inc.

Profit and Loss

January 2021 - December 2023

	JAN - DEC 2021	JAN - DEC 2022	JAN - DEC 2023	TOTAL
Income				
Donation	1,060,580.00	1,076,200.00	2,389,247.27	\$4,526,027.27
Interest	32.81	240.21	1,621.38	\$1,894.40
Total Income	\$1,060,612.81	\$1,076,440.21	\$2,390,868.65	\$4,527,921.67
GROSS PROFIT	\$1,060,612.81	\$1,076,440.21	\$2,390,868.65	\$4,527,921.67
Expenses				
Advertising & Marketing			2,874.61	\$2,874.61
Bank Charges & Fees			50.00	\$50.00
Community	52,883.88	58,726.93	95,580.94	\$207,191.75
Dues & subscriptions	5,875.00	22,037.00	13,451.00	\$41,363.00
Job Supplies			16,529.51	\$16,529.51
Legal & Professional Services	4,157.50	35,469.00	-3,660.87	\$35,965.63
Licenses	20.00	130.00	147.74	\$297.74
Office Supplies & Software	2,392.31	4,328.65	20,685.57	\$27,406.53
Payroll Expenses	2,074.00	773.00	1,005.77	\$3,852.77
Company Contributions				\$0.00
Retirement			21,541.33	\$21,541.33
Total Company Contributions			21,541.33	\$21,541.33
Taxes	20,139.30	29,096.88	56,909.54	\$106,145.72
Wages	252,230.62	369,365.38	631,373.58	\$1,252,969.58
Total Payroll Expenses	274,443.92	399,235.26	710,830.22	\$1,384,509.40
Postage	66.95		241.11	\$308.06
Software Development	275,050.39	391,143.37	632,668.69	\$1,298,862.45
Software Security Testing	15,400.00			\$15,400.00
Sponsorship	12,000.00	27,000.00	9,950.30	\$48,950.30
Theft/Scam		175,414.00		\$175,414.00
Travel			7,461.42	\$7,461.42
Website	57,574.85	173,881.76	565,080.51	\$796,537.12
Total Expenses	\$699,864.80	\$1,287,365.97	\$2,071,890.75	\$4,059,121.52
NET OPERATING INCOME	\$360,748.01	\$ -210,925.76	\$318,977.90	\$468,800.15
Other Income				
Other Miscellaneous Income	32,250.00			\$32,250.00
Total Other Income	\$32,250.00	\$0.00	\$0.00	\$32,250.00
Other Expenses				
Unrealized Gain or Loss	0.00	0.00	0.00	\$0.00
Total Other Expenses	\$0.00	\$0.00	\$0.00	\$0.00
NET OTHER INCOME	\$32,250.00	\$0.00	\$0.00	\$32,250.00
NET INCOME	\$392,998.01	\$ -210,925.76	\$318,977.90	\$501,050.15