Websites:

1. <https://techbullion.com/top-10-smart-contract-auditing-companies-in-2025-ensuring-blockchain-security/>
2. https://geekflare.com/cybersecurity/smart-contract-auditing-firms/

What are the best smart contract auditing companies?

1. **Quantstamp …4567**
2. **PeckShield..3**
3. **SlowMist …45678**
4. **Certik …456**
5. **Hacken …..6787**
6. **ConsenSys Diligence ….56789**
7. **Hashlock ….5678**
8. **QuillAudits …4**
9. **Sigma Prime..3**
10. **Bunzz Audit**
11. OpenZeppelin….56789
12. Trail of Bits…456
13. Runtime Verification
14. Halborn..345
15. Cyfrin ..345
16. Three Sigma1
17. Antier Solutions1
18. Softstack
19. ChainSecurity…4
20. Spearbit.
21. Code4rena.2
22. codeHawks.2
23. kudelski Security1
24. Zellic1
25. Dedaub
26. Nethermind Security
27. Cyberscope
28. Certora1
29. SmartDec
30. Solidified
31. Chainsulting
32. Least Authority
33. Mythx
34. Bunzz Audit
35. Blocksec
36. Callisto smart
37. Immune Bytes
38. Arbitrary Execution
39. Blaize

When collecting and choosing the best smart contract auditing companies, base your selection on these critical criteria derived from industry best practices and expert insights:

# Key Bases to Choose the Best Smart Contract Auditing Company

**1. Expertise and Experience**

* Ensure the company has deep knowledge of blockchain platforms and smart contract languages relevant to your project (e.g., Solidity, Vyper).
* Look for auditors experienced in identifying complex vulnerabilities like reentrancy, integer overflows, and access control issues.
* Prefer firms that have audited projects similar in scope and technology to yours[4](https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies)[5](https://cryptojobslist.com/blog/smart-contract-audit-companies)[9](https://www.ulam.io/blog/smart-contract-auditing-services-how-to-pick-the-best-provider).

**2. Reputation and Track Record**

* Check for a solid reputation supported by client reviews, testimonials, and case studies.
* Verify if they have audited high-profile or well-known blockchain projects.
* Consider industry recognition and memberships in blockchain or cybersecurity organizations[4](https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies)[5](https://cryptojobslist.com/blog/smart-contract-audit-companies)[6](https://www.webmobinfo.ch/blog/smart-contract-audit-cost-and-benefits)[9](https://www.ulam.io/blog/smart-contract-auditing-services-how-to-pick-the-best-provider).

**3. Audit Methodology and Scope**

* Understand their audit process: combination of manual code review, automated testing, fuzzing, gas optimization, and formal verification.
* Confirm whether they provide comprehensive coverage including security vulnerabilities, code correctness, and performance aspects.
* Ask if they offer competitive audits (crowdsourced) or dedicated private audits[4](https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies)[6](https://www.webmobinfo.ch/blog/smart-contract-audit-cost-and-benefits)[7](https://coinbureau.com/guides/how-to-audit-smart-contract/)[11](https://hacken.io/discover/smart-contract-audit-process/).

**4. Quality and Clarity of Reporting**

* The audit report should be detailed, clear, and actionable, outlining all vulnerabilities, their severity, and recommended fixes.
* Reports must be understandable to both developers and stakeholders to facilitate remediation and trust-building[6](https://www.webmobinfo.ch/blog/smart-contract-audit-cost-and-benefits)[7](https://coinbureau.com/guides/how-to-audit-smart-contract/)[11](https://hacken.io/discover/smart-contract-audit-process/).

**5. Customized Services and Flexibility**

* Choose auditors who tailor their services to your project’s specific needs and complexity.
* Pricing transparency and flexibility to fit your budget and timeline are important.
* Some firms offer follow-up audits and ongoing monitoring post-deployment, which adds value[5](https://cryptojobslist.com/blog/smart-contract-audit-companies)[6](https://www.webmobinfo.ch/blog/smart-contract-audit-cost-and-benefits)[9](https://www.ulam.io/blog/smart-contract-auditing-services-how-to-pick-the-best-provider).

**6. Technical Proficiency and Tools**

* The auditor should use a blend of automated tools (e.g., Mythril, Slither, Echidna) and manual reviews to maximize vulnerability detection.
* Preference for firms employing formal verification or AI-assisted auditing enhances assurance[4](https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies)[7](https://coinbureau.com/guides/how-to-audit-smart-contract/)[10](https://chain.link/education-hub/how-to-audit-smart-contract).

**7. Communication and Support**

* Effective and ongoing communication during the audit process is crucial.
* The auditor should be responsive, explain findings clearly, and provide guidance on fixing issues[6](https://www.webmobinfo.ch/blog/smart-contract-audit-cost-and-benefits)[7](https://coinbureau.com/guides/how-to-audit-smart-contract/)[11](https://hacken.io/discover/smart-contract-audit-process/).

**8. Cost and Time Considerations**

* Understand the audit fee structure relative to project complexity.
* Obtain multiple quotes to find a balance between cost and thoroughness.
* Avoid choosing solely on price; investing in a thorough audit reduces long-term risks[6](https://www.webmobinfo.ch/blog/smart-contract-audit-cost-and-benefits).

By evaluating smart contract auditing companies on these bases, you ensure selection of a partner capable of delivering high-quality, reliable security audits that protect your project and its users effectively.

**Citations:**

1. <https://www.rightfirms.co/directory/blockchain-technology/smart-contract-audit>
2. <https://www.alchemy.com/dapps/best/blockchain-auditing-companies>
3. <https://www.quillaudits.com/blog/smart-contract/top-smart-contract-audit-firms>
4. <https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies>
5. <https://cryptojobslist.com/blog/smart-contract-audit-companies>
6. <https://www.webmobinfo.ch/blog/smart-contract-audit-cost-and-benefits>
7. <https://coinbureau.com/guides/how-to-audit-smart-contract/>
8. <https://clutch.co/in/developers/blockchain/smart-contract-audit>
9. <https://www.ulam.io/blog/smart-contract-auditing-services-how-to-pick-the-best-provider>
10. <https://chain.link/education-hub/how-to-audit-smart-contract>
11. <https://hacken.io/discover/smart-contract-audit-process/>
12. <https://metana.io/blog/smart-contract-auditing-essential-guide-for-blockchain-security/>

Answer from Perplexity: [pplx.ai/share](https://www.perplexity.ai/search/pplx.ai/share)

# documentation style works well:

* **Company Overview:** Brief introduction highlighting the company’s focus and expertise.
* **Services Offered:** Clear list of key services such as auditing, monitoring, bug bounty management, and upgradeable security.
* **Audit Process:** Step-by-step explanation of how the audit is conducted, from client engagement to final reporting and re-audits.
* **Unique Selling Points:** Emphasizes what differentiates the company, such as manual auditing, tailored client focus, and ongoing support.
* **Client Collaboration:** Highlights interactive and transparent communication during the audit.
* **Additional Support:** Mentions post-audit services like incident response and continuous monitoring.
* **Professional Tone:** Uses precise, journalistic language suitable for business and technical audiences.

Here are the main key points you should cover when describing OpenZeppelin in the context of smart contract auditing:

**OpenZeppelin Smart Contract Auditing – Key Points**

**1. Company Overview and Expertise**

* OpenZeppelin is a leading blockchain security firm specializing in smart contract security and development.
* Known for their widely-used, secure, open-source smart contract libraries (e.g., ERC-20, ERC-721, ERC-1155).
* Provides comprehensive smart contract audits combining manual expert review and automated tools.
* Deep expertise in Ethereum Virtual Machine (EVM) environments and other EVM-compatible chains like Avalanche.
* Trusted by major projects including Bancor, Celo, 1inch, The Graph, and more[4](https://build.avax.network/integrations/openzeppelin)[7](https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies).

**2. Audit Process and Methodology**

* Multi-stage, thorough audit process tailored to each project’s needs.
* Initial preparation includes scope definition, codebase confirmation, and documentation review.
* At least two security auditors conduct a line-by-line manual code review to catch logical flaws, access control issues, and edge cases.
* Strategic use of automated tools such as static analysis and fuzz testing to identify common vulnerabilities.
* Continuous communication with clients, including early alerts for critical issues.
* Delivery of detailed, categorized reports with severity levels, remediation advice, and contextual explanations.
* One round of fix review to verify remediations before finalizing the audit report.
* Optional public release of audit reports to demonstrate transparency and build community trust[5](https://learn.openzeppelin.com/security-audits/readiness-guide)[6](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

**3. Security Features and Tools**

* OpenZeppelin Contracts are extensively tested and audited, providing a secure foundation for decentralized applications.
* Offers robust access control mechanisms, including role-based access control (RBAC).
* Provides OpenZeppelin Defender, a platform for automating and securing smart contract operations, including workflow automation and security monitoring.
* Encourages community involvement through forums and bug bounty programs to identify and resolve vulnerabilities proactively[4](https://build.avax.network/integrations/openzeppelin).

**4. Use Cases and Industry Impact**

* Audits are particularly suited for DeFi protocols, governance systems (DAOs), open-source projects, and any EVM-based smart contract deployments.
* Extensive experience auditing Layer 1 and Layer 2 solutions, NFT projects, oracles, and bridges.
* In 2024 alone, OpenZeppelin conducted 400 audits, identifying over 190 critical and high-severity issues, helping prevent costly exploits[6](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

**5. Customization and Integration**

* OpenZeppelin Contracts are designed for extensibility, allowing developers to add custom features while retaining security benefits.
* Supports integration with custom security solutions and risk management tools to tailor security to specific business needs.
* Encourages continuous security reviews and adherence to best practices to maintain contract integrity over time[4](https://build.avax.network/integrations/openzeppelin).

**6. Community and Educational Role**

* OpenZeppelin maintains a strong developer community and provides extensive educational resources.
* Shares knowledge gained from audits and research to improve ecosystem security.
* Promotes transparency by publishing audit reports and security advisories, fostering trust in blockchain projects[4](https://build.avax.network/integrations/openzeppelin)[6](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

These points collectively highlight OpenZeppelin’s comprehensive approach to smart contract auditing, their technical rigor, community engagement, and the practical benefits they bring to blockchain security.

**Citations:**

1. <https://www.openzeppelin.com/security-audits>
2. [https://www.openzeppelin.com](https://www.openzeppelin.com/)
3. <https://www.onesafe.io/blog/how-does-openzeppelin-transform-blockchain-security>
4. <https://build.avax.network/integrations/openzeppelin>
5. <https://learn.openzeppelin.com/security-audits/readiness-guide>
6. <https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits>
7. <https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies>
8. <https://diligence.consensys.io/blog/2023/04/how-to-prepare-for-a-smart-contract-audit-with-consensys-diligence/>
9. <https://www.openzeppelin.com/request>
10. [https://docs.openzeppelin.com](https://docs.openzeppelin.com/)
11. <https://forum.openzeppelin.com/t/smart-contract-vulnerability-audit-tool/34181>
12. <https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/audits/2017-03.md>
13. <https://blog.openzeppelin.com/security-audits/>
14. <https://forum.openzeppelin.com/t/my-coding-journey-from-finance-to-smart-contract-auditor/39251>
15. <https://www.openzeppelin.com/solidity-contracts>
16. <https://www.openzeppelin.com/readiness-guide>
17. <https://docs.openzeppelin.com/defender/module/audit>

When preparing documentation on OpenZeppelin's smart contract auditing services, the main key points to cover are:

**1. Overview of OpenZeppelin Auditing Approach**

* OpenZeppelin performs **comprehensive, multi-stage security audits** designed to identify and address vulnerabilities before deployment.
* Their audits combine **manual line-by-line code review** by at least two security researchers with **automated tools** such as static analysis and fuzz testing.
* The process includes reviewing the **system architecture, business logic, and access control mechanisms** to ensure thorough coverage.
* OpenZeppelin emphasizes a **collaborative approach**, engaging closely with the client’s developers throughout the audit to understand project-specific nuances and optimize security outcomes[2](https://www.openzeppelin.com/security-audits)[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

**2. Audit Process and Methodology**

* **Pre-Audit Preparation:** Reviewing project documentation, running test suites, and using proprietary tools like Code Inspector to detect low-severity issues early.
* **Scope Definition:** Confirming the exact codebase version and audit scope to avoid last-minute changes and focus efforts effectively.
* **Kick-Off & Planning:** Formal kickoff call to align on timelines, scope, and audit plan with assigned responsibilities.
* **Thorough Code Review:** Manual inspection of every line of code, checking for logical flaws, edge cases, and access control correctness.
* **Automated Testing:** Use of static analysis tools and fuzz testing to uncover vulnerabilities that manual review might miss.
* **Fix Review:** Clients submit fixes in dedicated pull requests; OpenZeppelin verifies and ensures no new issues are introduced.
* **Retro & Knowledge Sharing:** Post-audit documentation of lessons learned and updating internal knowledge bases to improve future audits.
* **Final Report Delivery:** Polished audit report with verified fixes, optionally published publicly for transparency[2](https://www.openzeppelin.com/security-audits)[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

**3. Expertise and Technology Focus**

* OpenZeppelin is a leader in **Solidity smart contract audits**, also providing specialized audits for **Cairo language** used in zk-Rollups and Layer 2 solutions.
* Their audits go beyond code review to include **system design, adherence to Ethereum Improvement Proposals (EIPs), and emerging blockchain technologies**.
* They maintain a large internal knowledge base and proprietary tools to enhance audit quality and efficiency[2](https://www.openzeppelin.com/security-audits)[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits)[6](https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies).

**4. Impact and Industry Standing**

* Trusted by major blockchain projects and protocols, OpenZeppelin has audited over **1 million lines of code** and uncovered **1000+ critical and high-severity vulnerabilities**.
* Their work has helped secure over **$50 billion in Total Value Locked (TVL)** across decentralized finance and other blockchain applications.
* OpenZeppelin also provides widely used **open-source contract libraries** that help developers build secure smart contracts from the ground up[5](https://www.openzeppelin.com/).

**5. Common Vulnerabilities Addressed**

* The audits focus on identifying common smart contract risks such as **reentrancy, integer overflows/underflows, improper access controls, and business logic errors**.
* They also assess project-specific custom logic and edge cases that could introduce unforeseen vulnerabilities[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

**6. Client Collaboration and Transparency**

* OpenZeppelin emphasizes transparency by offering clients the option to **publicly release audit reports**, fostering community trust.
* They provide detailed, actionable recommendations balancing best practices with project-specific needs to facilitate remediation and secure deployments[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

These points comprehensively cover OpenZeppelin’s smart contract auditing services, methodology, expertise, and industry impact, providing a clear, structured foundation for your documentation.

**1. Overview of OpenZeppelin Auditing Approach**

* OpenZeppelin performs **comprehensive, multi-stage security audits** designed to identify and address vulnerabilities before deployment.
* Their audits combine **manual line-by-line code review** by at least two security researchers with **automated tools** such as static analysis and fuzz testing.
* The process includes reviewing the **system architecture, business logic, and access control mechanisms** to ensure thorough coverage.
* OpenZeppelin emphasizes a **collaborative approach**, engaging closely with the client’s developers throughout the audit to understand project-specific nuances and optimize security outcomes[2](https://www.openzeppelin.com/security-audits)[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

**2. Audit Process and Methodology**

* **Pre-Audit Preparation:** Reviewing project documentation, running test suites, and using proprietary tools like Code Inspector to detect low-severity issues early.
* **Scope Definition:** Confirming the exact codebase version and audit scope to avoid last-minute changes and focus efforts effectively.
* **Kick-Off & Planning:** Formal kickoff call to align on timelines, scope, and audit plan with assigned responsibilities.
* **Thorough Code Review:** Manual inspection of every line of code, checking for logical flaws, edge cases, and access control correctness.
* **Automated Testing:** Use of static analysis tools and fuzz testing to uncover vulnerabilities that manual review might miss.
* **Fix Review:** Clients submit fixes in dedicated pull requests; OpenZeppelin verifies and ensures no new issues are introduced.
* **Retro & Knowledge Sharing:** Post-audit documentation of lessons learned and updating internal knowledge bases to improve future audits.
* **Final Report Delivery:** Polished audit report with verified fixes, optionally published publicly for transparency[2](https://www.openzeppelin.com/security-audits)[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

**3. Expertise and Technology Focus**

* OpenZeppelin is a leader in **Solidity smart contract audits**, also providing specialized audits for **Cairo language** used in zk-Rollups and Layer 2 solutions.
* Their audits go beyond code review to include **system design, adherence to Ethereum Improvement Proposals (EIPs), and emerging blockchain technologies**.
* They maintain a large internal knowledge base and proprietary tools to enhance audit quality and efficiency[2](https://www.openzeppelin.com/security-audits)[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits)[6](https://www.cyfrin.io/blog/top-10-smart-contract-auditing-companies).

**4. Impact and Industry Standing**

* Trusted by major blockchain projects and protocols, OpenZeppelin has audited over **1 million lines of code** and uncovered **1000+ critical and high-severity vulnerabilities**.
* Their work has helped secure over **$50 billion in Total Value Locked (TVL)** across decentralized finance and other blockchain applications.
* OpenZeppelin also provides widely used **open-source contract libraries** that help developers build secure smart contracts from the ground up[5](https://www.openzeppelin.com/).

**5. Common Vulnerabilities Addressed**

* The audits focus on identifying common smart contract risks such as **reentrancy, integer overflows/underflows, improper access controls, and business logic errors**.
* They also assess project-specific custom logic and edge cases that could introduce unforeseen vulnerabilities[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

**6. Client Collaboration and Transparency**

* OpenZeppelin emphasizes transparency by offering clients the option to **publicly release audit reports**, fostering community trust.
* They provide detailed, actionable recommendations balancing best practices with project-specific needs to facilitate remediation and secure deployments[4](https://blog.openzeppelin.com/what-is-a-smart-contract-audit-lessons-from-openzeppelins-1000-audits).

These points comprehensively cover OpenZeppelin’s smart contract auditing services, methodology, expertise, and industry impact, providing a clear, structured foundation for your documentation.

What to Cover in Smart Contract Audit Documentation

When preparing smart contract audit documentation, especially for a company like OpenZeppelin, the key points to include are:

1. **Introduction and Scope**

* Overview of the project and smart contracts audited.
* The purpose and scope of the audit: which contracts, versions, and functionalities were reviewed.
* Any dependencies or third-party libraries involved (e.g., OpenZeppelin libraries).
* Disclaimer clarifying the audit’s limits and that it does not guarantee absolute security.

2. **Audit Methodology and Approach**

* Description of the audit process combining **manual code review** and **automated testing** tools (e.g., static analysis, fuzz testing).
* Explanation of how the team prepared for the audit, including reviewing documentation and running test suites.
* Details on collaboration with the project team during the audit.
* Use of proprietary tools such as OpenZeppelin’s Code Inspector to detect low-severity issues early.
* Classification of vulnerabilities by severity (Critical, High, Medium, Low, Informational).

3. **Findings and Issues**

* Detailed list of identified vulnerabilities and issues.
* For each issue: title, description, severity, potential impact, and affected code areas.
* Status of each issue (Unresolved, Resolved, Partially Resolved, etc.).
* Clear explanations to make technical findings understandable to different stakeholders.

4. **Recommendations and Remediation**

* Practical, prioritized recommendations for fixing each identified issue.
* Best practices tailored to the project’s specific needs and architecture.
* Guidance on secure coding standards and architectural improvements.

5. **Fix Review Process**

* Description of how fixes were reviewed and verified by auditors.
* Interaction and communication with developers during remediation.
* Confirmation that fixes do not introduce new vulnerabilities.

6. **Summary and Conclusion**

* Overall assessment of the contract’s security and readiness for deployment.
* Any residual risks or areas requiring ongoing attention.
* Suggestions for continuous security monitoring post-deployment.

7. **Appendices and Supplementary Information**

* Tools and automated scanners used during the audit (e.g., Slither, Echidna).
* Test coverage details and unit testing performed.
* References to relevant standards or vulnerability registries (e.g., SWC Registry).
* Glossary or explanations of technical terms for non-technical readers.

Additional Considerations for OpenZeppelin Audit Documentation

* Emphasize the **client-centered approach** OpenZeppelin follows, including ongoing communication and potential for long-term collaboration.
* Highlight the use of the **Defender platform** for issue tracking and communication during the audit and fix review phases.
* Include a note on the importance of **code freeze** before auditing to ensure the reviewed code matches the deployed version.
* Clarify that the audit report can be made **public** to increase transparency and community trust.
* Stress the importance of **input validation, dependency checks, and adherence to secure development lifecycle phases** as part of audit preparation.
* I want to prepare a documentation for the best smart contract auditing company ,what are the main topics I can cover in the documentation about the company

## **Company Overview**

Core Services and Expertise

Audit Methodology and Process

Notable Achievements and Impact

Common Vulnerabilities Addressed

Tools and Technologies Used

Reporting and Documentation

Post-Audit Support and Monitoring

Why Choose This Company?

Certainly! Below is a **well-structured documentation template** for smart contract auditing companies, synthesizing your attached files and best practices. This template is suitable for presenting to stakeholders, clients, or internal teams.

**🛡️ Smart Contract Auditing Companies – Comprehensive Documentation**

This document presents an in-depth overview of top-tier smart contract auditing companies in the blockchain ecosystem. Each profile includes core company information, service offerings, audit methodologies, tooling, client ecosystems, and contact details.

In addition to company fundamentals, this guide highlights each firm’s unique strengths, security philosophies, and contributions to the Web3 security landscape. Where available, sample audit reports and notable client projects are referenced to demonstrate real-world impact and experience. The documentation also outlines how to engage with each company for audit requests and provides links to educational resources, community channels, and technical tooling.

Whether you are a blockchain developer, protocol founder, or security researcher, this comprehensive resource is designed to help you evaluate, compare, and connect with the leading smart contract auditing firms shaping the future of decentralized security.

**🔹 1. CertiK**

**🏢 Company Overview**

* **Founded:** 2018, by Yale and Columbia professors
* **Mission:** Secure the Web3 world with formal verification and AI-driven technology
* **Vision:** To become the default standard for blockchain security
* **Positioning:** Industry leader; audited 3,800+ projects with $360B+ in assets secured

**🛡️ Core Services**

* **Smart Contract Auditing:** Manual + AI static analysis, formal verification
* **Security Consulting:** Protocol architecture, governance risk
* **Post-Audit Monitoring:** Skynet - real-time risk engine

**📊 Audit Methodology**

* AI + manual review
* Tools: CertiK’s proprietary engine, Slither, Manticore
* Issue levels: Critical, Major, Medium, Minor, Info
* Patch validation via GitHub

**🔐 Security Philosophy**

* Formal methods > trust assumptions
* AI-enhanced behavioral anomaly detection

**🧾 Sample Audit Reports**

* PancakeSwap: <https://www.certik.com/projects/pancakeswap>
* ShibaSwap, Aave, dYdX, and many more

**🌍 Clients & Ecosystem**

* Binance, Polygon, OKX, Terra, ApeCoin DAO
* Multichain, Solana, BSC projects

**⚙️ Tooling & Infrastructure**

* CertiK Skynet, DeepSEA formal tools
* Bug bounty platform: CertiKShield

**📦 How to Request an Audit**

* Website: [certik.com](https://www.certik.com/)
* Intake form: <https://www.certik.com/request-audit>

**📚 Learning & Community**

* YouTube, Blog, Exploits Library

**📞 Contact Info**

* Email: [contact@certik.org](mailto:contact@certik.org)
* Telegram: @certikorg

**🔹 2. OpenZeppelin**

**🏢 Company Overview**

* **Founded:** 2015
* **Mission:** Build secure, open-source infrastructure for Ethereum
* **Vision:** Empower builders with security-first tools and practices
* **Positioning:** Ethereum core contributor; $50B+ assets secured

**🛡️ Core Services**

* **Smart Contract Auditing:** Deep manual code review, EVM expertise
* **Security Consulting:** Threat modeling, secure protocol design
* **Post-Audit Monitoring:** OpenZeppelin Defender: relayers, sentinels

**📊 Audit Methodology**

* Manual-first + Slither, MythX, Echidna, custom scripts
* Formal specs for upgradeable contracts

**🔐 Security Philosophy**

* Open-source transparency
* Checks-Effects-Interactions pattern, modular design

**🧾 Sample Audit Reports**

* Aave, Uniswap V3, Compound
* <https://blog.openzeppelin.com/security-audits/>

**🌍 Clients & Ecosystem**

* Ethereum Foundation, Arbitrum, Optimism, Base (Coinbase)

**⚙️ Tooling & Infrastructure**

* OpenZeppelin Contracts library, Defender, Wizard

**📦 How to Request an Audit**

* GitHub + Contact form via openzeppelin.com

**📚 Learning & Community**

* Forum, blog, tutorials, ERC spec contributors

**📞 Contact Info**

* Email: [contact@openzeppelin.com](mailto:contact@openzeppelin.com)
* Forum: forum.openzeppelin.com

**🔹 3. ConsenSys Diligence**

**🏢 Company Overview**

* **Founded:** Part of ConsenSys (founded by Joseph Lubin, Ethereum co-founder)
* **Mission:** Strengthen Ethereum ecosystem via expert audits
* **Vision:** Formal security practices for decentralized systems
* **Positioning:** Ethereum-native auditors

**🛡️ Core Services**

* **Smart Contract Auditing:** Threat modeling, manual + symbolic analysis
* **Security Consulting:** Secure protocol workflows, multi-sig setup

**📊 Audit Methodology**

* Tools: Scribble, MythX, Diligence Fuzzing
* Focused on formal specs + logic correctness

**🔐 Security Philosophy**

* Developer-first security integration
* Push toward verifiable correctness

**🧾 Sample Audit Reports**

* MakerDAO, Metamask, 1inch
* <https://consensys.net/diligence/audits/>

**🌍 Clients & Ecosystem**

* Ethereum mainnet protocols, L2s, DAOs

**⚙️ Tooling & Infrastructure**

* MythX, Scribble, Harvey

**📦 How to Request an Audit**

* <https://consensys.net/diligence/audit-request>

**📚 Learning & Community**

* Diligence Blog, Solidity Friday newsletters

**📞 Contact Info**

* [audits@consensys.net](mailto:audits@consensys.net)

**🔹 4. Trail of Bits**

**🏢 Company Overview**

* **Founded:** 2012
* **Mission:** Secure mission-critical software (Web3, defense, enterprise)
* **Vision:** Blend formal methods with battle-tested security engineering
* **Positioning:** Elite security firm trusted by top DeFi protocols

**🛡️ Core Services**

* **Smart Contract Auditing:** Manual & symbolic execution + fuzzing
* **Security Consulting:** Secure compiler/toolchain design

**📊 Audit Methodology**

* Tools: Slither (they built it), Manticore, Echidna
* Advanced formal analysis + test generators

**🔐 Security Philosophy**

* White-box by default
* Attack simulation focus

**🧾 Sample Audit Reports**

* Compound, Balancer, Element Finance
* <https://github.com/trailofbits/publications>

**🌍 Clients & Ecosystem**

* Ethereum Foundation, OpenSea, Coinbase

**⚙️ Tooling & Infrastructure**

* Slither, Echidna, Manticore (open source)

**📦 How to Request an Audit**

* [trailofbits.com](https://www.trailofbits.com/services/smart-contract-audits)

**📚 Learning & Community**

* GitHub, blog, whitepapers

**📞 Contact Info**

* [info@trailofbits.com](mailto:info@trailofbits.com)

**🔹 5. Hacken**

**🏢 Company Overview**

* **Founded:** 2017, Ukraine-based
* **Mission:** Web3 cybersecurity ecosystem
* **Vision:** End-to-end security infrastructure for DeFi, exchanges
* **Positioning:** Full-stack security (audit + bug bounty + KYC)

**🛡️ Core Services**

* **Smart Contract Auditing:** Manual + automated + test-driven
* **Security Consulting:** Exchange security, blockchain protocol audits
* **Post-Audit Monitoring:** Real-time threat detection, AML, bug bounty

**📊 Audit Methodology**

* OWASP-based risk classification
* Slither, Foundry, Hardhat testing
* Tools: HackenProof platform

**🔐 Security Philosophy**

* Proactive security lifecycle
* Bug bounty + public transparency

**🧾 Sample Audit Reports**

* VeChain, Solana, 1inch, Avalanche projects

**🌍 Clients & Ecosystem**

* KuCoin, 1inch, Gate.io, MEXC

**⚙️ Tooling & Infrastructure**

* HackenProof, AML integrations

**📦 How to Request an Audit**

* [hacken.io/security-audit](https://hacken.io/)

**📚 Learning & Community**

* Hacken Twitter, Blog, Proof of Hack reports

**📞 Contact Info**

* [audit@hacken.io](mailto:audit@hacken.io)
* Telegram: @hacken\_io

**🔹 6. QuillAudits**

**🏢 Company Overview**

* **Founded:** 2018, India-based
* **Mission:** Secure multi-chain Web3 ecosystems through scalable audits
* **Vision:** Democratize Web3 security across chains
* **Positioning:** Audited 1400+ projects across 20+ blockchains

**🛡️ Core Services**

* **Smart Contract Auditing:** Manual + static analysis, fuzzing, symbolic testing
* **Security Consulting:** Threat modeling, DAO governance

**📊 Audit Methodology**

* Multi-tool stack: Slither, MythX, Echidna
* Automated + manual hybrid audits

**🔐 Security Philosophy**

* Emphasizes human review + chain-specific attack vectors

**🧾 Sample Audit Reports**

* Polygon, Avalanche, Near ecosystem
* <https://audits.quillhash.com/>

**🌍 Clients & Ecosystem**

* Klaytn, XDC, Persistence, NFT marketplaces

**⚙️ Tooling & Infrastructure**

* QuillMonitor, fuzz testing framework

**📦 How to Request an Audit**

* <https://audits.quillhash.com/#get-started>

**📚 Learning & Community**

* Security blog, chain-specific insights

**📞 Contact Info**

* [audits@quillhash.com](mailto:audits@quillhash.com)
* Telegram: @quillhash

*(Continue with other companies such as Hashlock, Cyfrin, Spearbit, SlowMist, Zellic, PeckShield, etc., following the same structure.)*

**End of Documentation**

**Tips:**

* For each company, keep the structure consistent.
* Add or update sample audit reports and client lists as needed.
* For even more value, consider adding a section on "Roles and Responsibilities" and "Why Choose This Company?" for each profile.

If you need the documentation in a downloadable format or want to continue with more companies, let me know!

**Citations:**

1. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/13203368/4932a778-d036-47fd-81d6-156bbba7f196/paste.txt>
2. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/13203368/0f5da611-7160-496c-800e-014dc424d27e/paste-2.txt>
3. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/13203368/4f16598d-cc87-4486-a515-5f82cb55fbb2/paste-3.txt>
4. <https://ppl-ai-file-upload.s3.amazonaws.com/web/direct-files/attachments/13203368/ce3db133-8a44-4bee-9b40-53a3329a3919/paste-4.txt>

Answer from Perplexity: [pplx.ai/share](https://www.perplexity.ai/search/pplx.ai/share)