Cyberscope

Audit Report

Elar

October 2023

Audited by © cyberscope

Table of Contents

Overview

Elar is a robust replicated state machine designed to uphold a secure, local copy of a cryptographic ledger. It effectively facilitates the processing of transactions while ensuring consensus among a designated group of peers. Elar's foundation is rooted in the implementation of the Stellar Consensus Protocol, a federated consensus protocol renowned for its reliability. Elar is built using C++14 and boasts cross-platform compatibility, making it compatible with Linux, OSX, and Windows operating systems.

Architecture

In our code audit, we've examined the process-level architecture of the Elar system. The application consists of several key components, each serving a distinct role in the system's operation. This comprehensive architecture encompasses components responsible for ledger management, peer-to-peer communication, transaction synchronization, transaction processing, cryptographic operations, and database management. Two specific components that may appear slightly obscure in naming are "BucketList" and "SCP," each with its unique purpose and significance within the system.

Our review of the process-level architecture of the Elar system underscores the significance of each component in contributing to the system's overall functionality and reliability. The clear separation of responsibilities among these components, along with the unique roles played by "BucketList" and "SCP," demonstrates a well-thought-out and organized architecture designed to meet the demands of a complex distributed system.

Review of Network-Level

In our code audit, we've analyzed the network-level architecture of the Elar system, particularly focusing on the role and design considerations of validators within the network. The overarching philosophy guiding the architecture is to maintain simplicity within validators while offloading specific responsibilities to other parts of the system.

The network-level architecture of Elar emphasizes simplicity and efficient resource utilization. Validators are designed with a specific focus on offloading responsibilities to other parts of the system that are better suited for specific tasks. By maintaining an even system-load profile and prioritizing statelessness, the architecture seeks to provide a stable and reliable network environment. These architectural principles align with the goals of scalability, performance, and robustness within a distributed system.

Components

Certainly, here's a rephrased version of the information about the major components of the Elar system without numeric ordering:

Herder

Herder plays a pivotal role in bridging SCP with the broader stellar-core system. It offers concrete implementations of essential methods used by SCP to interact with peers, compare values, validate signatures, and more. Herder often delegates tasks to other components as needed. Detailed insights can be found in the src/herder/readme.md documentation.

Overlay

Overlay is responsible for establishing and maintaining connections with known peers and managing message propagation within the network. It efficiently disseminates messages and retrieves necessary data from peers to facilitate consensus. Additional details can be explored in the src/overlay/readme.md section. Other data downloads are handled separately, as explained in ./architecture.md.

Ledger

The Ledger component is in charge of applying transaction sets externalized by SCP. It also relays externalization events to various system components, such as updating ledger entries, triggering history data publishing, and notifying the overlay system about message flooding. The src/ledger/readme.md provides comprehensive information on its functionality.

History

History serves as the data publisher, archiving transaction and ledger entries to off-site permanent storage for auditing purposes and as a source of catch-up data for other nodes. When a node falls behind, the history system retrieves catch-up data and submits it to the

Ledger component, following a two-step process: first for security verification and then for application. Refer to src/history/readme.md for further insights.

BucketList

BucketList is responsible for organizing ledger entries on disk, optimizing them for hashing and block-catch-up processes. It coordinates the hashing and deduplication of buckets through multiple background threads. Explore src/bucket/readme.md for detailed information on its role and operations.

Transactions

Transactions encompass the various transaction types within the system. Detailed documentation on the implementation of different transaction types can be found in the src/transactions/readme.md section.

These component descriptions provide an overview of the major building blocks of the Elar system, each contributing to its functionality and overall operation. Further documentation within each component's respective readme.md directory can offer more in-depth insights into their specific functionalities and interactions within the system.

Stellar Consensus Protocol (SCP)

The SCP component serves as our tailored implementation of the Stellar Consensus Protocol. It operates independently from the rest of the system, handling candidate black-box values and signaling consensus achievements across the network. For a detailed explanation, please refer to the documentation in the src/scp/readme.md directory.

Review of the Stellar Consensus Protocol (SCP) Subsystem

In our code review of the Elar system, we have evaluated the SCP subsystem—an abstract implementation of the Stellar Consensus Protocol (SCP). SCP is a federated Byzantine agreement protocol, integral to the system's functionality. Notably, SCP is designed to drive a distributed system based on the concept of a "replicated state machine." It's worth mentioning that SCP is intentionally designed without tying it to specific concepts like "slot" or "value," allowing for flexibility in its application across different contexts.

Benefits of Subsystem Separation

One notable aspect we've observed is the deliberate separation of the SCP subsystem from the rest of the system. This architectural choice serves several valuable purposes. Firstly, it simplifies the modeling of SCP, making it more manageable to compare with the original protocol paper, facilitating auditability, and ensuring correctness. Moreover, this separation sets the stage for potential future extraction and reuse of SCP in various program contexts—a forward-looking strategy that enhances code versatility.

Implementation of Consensus with SCP

Our review revealed that to implement consensus using SCP, modules within the system are expected to create subclasses of the SCPDriver class. These subclasses must provide concrete implementations for abstract methods responsible for handling SCP-generated events. The SCPDriver class serves as the primary interface for receiving incoming messages related to SCP consensus. This modular approach facilitates code maintainability and extensibility.

Protocol Definitions and XDR

In our assessment, we've also noted that the SCP protocol messages are formally defined in XDR format within the Stellar-SCP.x file. However, for practical development purposes, much of the focus tends to be on modifying Stellar-types.x. This file houses essential base

types used throughout the implementation, such as hash and NodeID, which represent a node's identity.

Herder in stellar-core

Within the stellar-core program, we found a significant component known as Herder. Herder serves as the sole subclass of SCPDriver in this context, providing a specific interpretation of "slot" and "value." It plays a crucial role in connecting SCP with a designated broadcast communication medium (Overlay) and a specific replicated state machine (LedgerManager). Our assessment found Herder to be a key bridge in the system's architecture, facilitating SCP's integration with other core components.

For Further Insight

For a comprehensive understanding of the SCP protocol itself, we recommend referring to the original SCP paper. It provides extensive insights into the protocol's design and operation, serving as a valuable reference point for developers and auditors.

In conclusion, the separation and abstraction strategies employed within the SCP subsystem contribute significantly to the codebase's modularity, auditability, and potential for future reuse—a commendable approach that aligns well with best practices in code design and architecture.

Herder - Concrete Implementation of SCP

In our code review, we have closely examined the Herder component, specifically the HerderSCPDriver. This is a significant part of the Elar system, providing a concrete implementation of the Stellar Consensus Protocol (SCP). Unlike the abstract SCP protocol, HerderSCPDriver operates in terms of concepts familiar to Stellar developers, such as "transaction sets" and "ledger numbers.

Our assessment of HerderSCPDriver highlights its crucial role in making SCP actionable and relevant within the Elar system. By serving as a subclass of SCPDriver and implementing SCP in concrete terms, HerderSCPDriver plays a pivotal role in the overall functionality of Elar. It transforms SCP from an abstract concept into a practical and functional component within the system, enabling it to operate effectively in the context of "transaction sets" and "ledger numbers" as per Stellar's terminology. This alignment with real-world Stellar requirements is a key factor in the success of the Herder component within Elar.

Ledger

In our code audit, we've examined the concept of a ledger in the Elar system, which represents the state of the Stellar universe at a specific point in time. Here, we outline the key aspects of ledger management:

- Genesis Ledger
- Consensus and Transaction Sets
- Cryptographic Linkage and Ledger Chain
- Sequence Number Definition

The sequence number of a ledger is defined recursively based on its position in the ledger chain:

The genesis ledger is assigned a sequence number of 1.

Subsequent ledgers are assigned sequence numbers incrementally, starting from the genesis ledger. A ledger directly derived from a ledger with sequence number "n" is assigned a sequence number of "n+1."

This ledger management system provides a clear and secure mechanism for tracking the state of the Stellar universe over time. The use of cryptographic linkage and sequence numbers ensures the integrity of the ledger history, while SCP consensus rounds determine the application of transaction sets, allowing the universe to evolve and progress from one ledger to the next.

Historical Records Management

In our code audit, we've explored the management of historical records within the Elar system. Here's an overview of how historical records are handled

Responsibility of the History Module

The History module is primarily responsible for interfacing with "history archives." These archives are loosely defined as storage locations where flat files containing historical records can be stored and retrieved.

User-Defined History Archives

Users have the flexibility to define history archives by adding sections of the form [HISTORY.name] to the configuration file. Each history archive is identified by a name, which serves as a symbolic logging prefix. Multiple history archives can coexist within a single configuration file.

Configuring Get and Put Commands

For each history archive defined in the configuration, users can specify "get" and "put" commands. These commands define subprocesses that the stellar-core will execute to either deposit historical records into the archive or retrieve records from it. The commands are specified in the form of string templates, with placeholders {0} and {1} representing the files to be transmitted or received.

History Generation and Multiple Peers

Historical records are generated as part of the operation of the LedgerManager. It's important to note that multiple peers within the Elar network can publish history to multiple archives. This decentralization ensures redundancy and robustness in historical record storage.

Catchup Mode

In cases where the LedgerManager detects that it has fallen out of synchronization with its peers, it enters a "catchup mode." During this mode, the history module downloads and replays historical records from a history archive to the LedgerManager. This process aims to bring the LedgerManager back into synchronization with its peers.

This approach to managing historical records ensures the integrity and availability of historical data within the Elar system. By allowing users to configure history archives and define get/put commands, Elar offers flexibility in historical record management. The catchup mode is a critical feature that helps maintain synchronization with network peers, ensuring the consistency and reliability of historical data.

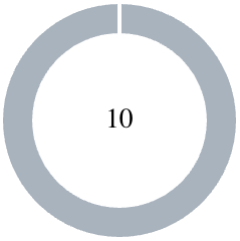
Buckets and the BucketList

In our code audit, we've explored the concept of buckets and the BucketList within the Elar system. These elements play a critical role in managing ledger entries efficiently. Here's a breakdown of how they function:

- Handling Large Sets of Ledger Entries
- Introduction of the BucketList
- BucketList Hash
- Checkpointing to History Storage
- Fast Catchup and Delta Application

The introduction of the BucketList and its hierarchical structure addresses the challenges of handling large sets of ledger entries efficiently. It enables the system to calculate cryptographic hashes, transmit minimal changes to peers during catchup, and maintain historical records for reference. This architectural approach enhances the integrity and performance of ledger management within the Elar system.

Findings Breakdown



Critical	0
Medium	0
Minor / Informative	10

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	10	0	0	0

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	UCWNV	Unsigned Comparison with Negative Values	Unresolved
●	EEC	Enhancing Explicit Constructors	Unresolved
●	LCPC	Legacy C-Style Pointer Casting	Unresolved
●	PPBCR	Parameter Passed by Const Reference	Unresolved
●	DE	Duplicate Expressions	Unresolved
●	LVS	Local Variable Shadowing	Unresolved
●	ITOFC	Invariant True or False Conditions	Unresolved
●	OVS	Optimizing Variable Scope	Unresolved
●	ECR	Enhancing Const References	Unresolved
●	USA	Using STL algorithms	Unresolved

UCWNV - Unsigned Comparison with Negative Values

Criticality	Minor / Informative
Location	overlay/TCPPeer.cpp#L568
Status	Unresolved

Description

The codebase contains unsigned expressions that are being compared with negative values. Such comparisons can lead to unexpected and erroneous behavior due to the nature of unsigned data types, which cannot represent negative values.

```
if (length <= 0 ||
```

Recommendation

To address this finding, it is advisable to review the identified comparisons and ensure that they are appropriate for the intended logic. If the intent is to check for values less than zero, consider using a signed data type instead of unsigned.

EEC - Enhancing Explicit Constructors

Criticality	Minor / Informative
Location	bucket/BucketApplicator.h#L66bucket/BucketInputIterator.h#L49bucket/BucketList.h#L363,L391bucket/BucketManagerImpl.h#L95catchup/ApplyBufferedLedgersWork.h#L19catchup/CatchupManagerImpl.h#L56crypto/ByteSlice.h#L59,L63,L66,L70,L74crypto/CryptoError.h#L16database/Database.cpp#L139database/Database.h#L40,L107herder/Upgrades.h#L40herder/HerderImpl.h#L62,L297herder/HerderPersistenceImpl.h#L17herder/HerderSCPDriver.h#L158herder/QuorumTracker.h#L50herder/SurgePricingUtils.h#L112herder/TransactionQueue.cpp#L1152herder/TxQueueLimiter.h#L19herder/TxSetUtils.h#L19herder/TxSetFrame.cpp#L174history/FileTransferInfo.h#L31history/HistoryManagerImpl.h#L46historywork/GetHistoryArchiveStateWork.h#L29invariant/InvariantManagerImpl.h#L34ledger/InMemoryLedgerTxnRoot.h#L35ledger/InternalLedgerEntry.h#L73,L142ledger/LedgerCloseMetaFrame.h#L18ledger/LedgerManagerImpl.h#L113ledger/LedgerTxn.h#L350ledger/TrustLineWrapper.cpp#L21,L379main/ExternalQueue.h#L17main/CommandHandler.h#L32main/CommandLine.cpp#L114,L122,L130main/PersistentState.h#L18overlay/BanManagerImpl.h#L21overlay/Floodgate.h#L53overlay/FlowControl.h#L87overlay/Peer.h#L120overlay/PeerAuth.h#L39overlay/PeerDoor.h#L35overlay/OverlayManagerImpl.h#L132overlay/OverlayMetrics.h#L24overlay/SurveyManager.h#L26overlay/TxAdvertQueue.h#L30scp/BallotProtocol.h#L99scp/NominationProtocol.h#L88simulation/CoreTests.cpp#L476simulation/LoadGenerator.h#L78,L105simulation/Simulation.h#L127simulation/Simulation.cpp#L714test/FuzzerImpl.cpp#L344,L1012,L1041test/SimpleTestReporter.h#L14test/TestMarket.h#L65test/TestUtils.h#L38,L55test/TxTests.h#L32,L33,L34,L35,L38transactions/BumpFootprintExpirationOpFrame.cpp#L17transactions/CreatePassiveSellOfferOpFrame.h#L14transactions/InvokeHostFunctionOpFrame.cpp#L182transactions/RestoreFootprintOpFrame.cpp#L18transactions/TransactionMetaFrame.h#L17transactions/TransactionSQL.cpp#L31util/BitSet.h#L122,L126util/LogSlowExecution.h#L22util/Timer.h#L182,L253,L254,L282util/TmpDir.h#L17,L29util/TxResource.h#L36,L46util/XDRStream.h#L39
Status	Unresolved

Description

The codebase contians instances where a class has a constructor with a single argument that lacks the 'explicit' keyword. An implicit single-argument constructor can lead to unintended type conversions and potentially introduce ambiguity in code behavior.

```
ByteSlice(xdr::opaque_array<N> const& arr)
ByteSlice(xdr::msg_ptr const& p) : mData(p->data()), mSize(p->size())
ByteSlice(std::vector<uint8_t> const& bytes)
ByteSlice(std::string const& bytes)
ByteSlice(char const* str) : ByteSlice((void const*)str, strlen(str))
```

Recommendation

To address this finding, it is recommended to review and update the constructor by adding the 'explicit' keyword to make the constructor's behavior more explicit. Doing so prevents unintended implicit type conversions and enhances code clarity.

LCPC - Legacy C-Style Pointer Casting

Criticality	Minor / Informative
Location	crypto/ByteSlice.h#L74crypto/StrKey.cpp#L26,L67ledger/LedgerTxnConfigSettingSQL.cpp#L118ledger/LedgerTxnContractCodeSQL.cpp#L136ledger/LedgerTxnContractDataSQL.cpp#L170,L172,L174ledger/LedgerTxnOfferSQL.cpp#L803process/ProcessManagerImpl.cpp#L791
Status	Unresolved

Description

The codebase is using C-style pointer casting. C-style pointer casting is a legacy practice that can introduce potential type-related issues and compatibility concerns. To improve code safety and maintainability, it is recommended to replace C-style pointer casting with modern C++ casting mechanisms.

```
ByteSlice(char const* str) : ByteSlice((void const*)str, strlen(str))
```

Recommendation

To address this finding, developers should refactor the code to utilize safer alternatives, such as static_cast, dynamic_cast, or reinterpret_cast, as appropriate for the specific use case. These modern C++ casting operators provide better type-checking and are less error-prone than C-style casting.

Replacing C-style pointer casting with modern C++ casting operators aligns the code with contemporary best practices, enhances code safety, and reduces the likelihood of type-related bugs. It also promotes code readability and maintainability by making the type conversions more explicit and comprehensible.

PPBCR - Parameter Passed by Const Reference

Criticality	Minor / Informative
Location	main/CommandLine.cpp#L448main/Config.cpp#L1717main/PersistentState.cpp#L38simulation/Simulation.cpp#L736util/FileSystemException.h#L25util/FileSystemException.cpp#L35util/TxResource.h#L36
Status	Unresolved

Description

The codebase contains function parameters that are passed by value when they could be more efficiently passed by const reference. When a parameter is passed by value, a copy of the argument is created, potentially leading to unnecessary memory consumption and performance overhead.

```
Simulation::metricsSummary(string domain)
```

Recommendation

To address this finding, it is recommended to review the identified function parameters and update their declarations to accept const references where appropriate. This change not only conserves memory but also improves the efficiency of function calls, especially when dealing with complex or large data structures.

DE - Duplicate Expressions

Criticality	Minor / Informative
Location	herder/test/HerderTests.cpp:4926
Status	Unresolved

Description

This finding identifies instances within the code where the same expression or sub-expression is duplicated on both sides of a logical or relational operator. Duplicate expressions can lead to redundant calculations, negatively impacting code efficiency and maintainability.

```
auto lastDexOpsBroadcasted = dexOpsBroadcasted;
auto lastNonDexOpsBroadcasted = nonDexOpsBroadcasted;
simulation->crankForAtLeast(broadcastPeriod + delta, false);
auto dexOpsPerPeriod = dexOpsBroadcasted - lastDexOpsBroadcasted;
```

Recommendation

To address this finding, it is advisable to review and refactor the code to eliminate the duplication of expressions. This can be achieved by:

- Storing the result of the common expression in a variable and then using that variable on both sides of the operator.
- Ensuring that the expression is only evaluated once, improving code performance and reducing the likelihood of errors.

By eliminating duplicate expressions, the code becomes more concise, easier to understand, and less prone to inconsistencies or bugs introduced by unintentional changes to one of the duplicated expressions.

LVS - Local Variable Shadowing

Criticality	Minor / Informative
Location	herder/TxSetFrame.cpp#L393,L415herder/TxSetFrame.h#L141
Status	Unresolved

Description

This finding points out instances within the codebase where a local variable is overshadowing a variable from an outer function. Local variable shadowing occurs when a new variable declared within a nested scope shares the same name as a variable in an enclosing scope.

```
size_t encodedSize = xdr::xdr_argpack_size(xdrTxSet);
```

Recommendation

To address this finding, it is recommended to review and modify the variable naming within the affected scope to prevent shadowing. This can be accomplished by:

- Renaming the local variable to give it a distinct and meaningful name.
- Ensuring that the variable's name does not conflict with variables in outer functions.
- Maintaining clear and unambiguous variable naming conventions to enhance code readability and maintainability.

Resolving local variable shadowing issues helps prevent unintended behavior, potential bugs, and enhances the clarity of variable scoping in the codebase.

ITOFC - Invariant True or False Conditions

Criticality	Minor / Informative
Location	bucket/Bucket.cpp#L766,L781
Status	Unresolved

Description

The codenbase contians specific conditions that are identified as either perpetually true or perpetually false based on the context in which they are evaluated. Recognizing such invariant conditions is crucial as they can lead to suboptimal code execution or indicate unintended logic.

```
if (!ni || (oi && ni && !refersToSameEntry(key(*oi), key(*ni)) && cmp(*oi,
*ni)))
...
```

Recommendation

To address this finding, a thorough review of the identified conditions should be conducted to:

- Verify the correctness of the code logic.
- Ensure that the conditions align with the intended behavior.
- Modify or optimize the code as necessary to improve its efficiency or accuracy.

Identifying and addressing invariant conditions contributes to the overall code quality and correctness, enhancing the robustness and reliability of the software.

OVS - Optimizing Variable Scope

Criticality	Minor / Informative
Location	herder/HerderPersistenceImpl.cpp#306,333history/StateSnapshot.cpp#67ledger/LedgerTxn.cpp#2822main/ApplicationUtils.cpp#51overlay/BanManagerImpl.cpp#97overlay/PeerManager.cpp#190,584scp/NominationProtocol.cpp#92transactions/LiquidityPoolDepositOpFrame.cpp#104,105util/Scheduler.cpp#368
Status	Unresolved

Description

Within the codebase, there are instances where variable scopes can be refined to enhance code clarity and maintainability. The primary recommendation is to minimize the scope of variables to the narrowest context where they are needed, reducing the risk of unintended side effects and improving code readability.

```
auto nodeIDString = KeyUtils::toStrKey(nodeID);
{
    ...
    st.exchange(soci::use(nodeIDString));
    ...
}
...
std::string qset64, qSetHashHex;

qSetHashHex = binToHex(qSetHash);
...
{
    ZoneNamedN(selectSCPQuorumsZone, "select scpquorums", true);
    soci::statement st = (sess.prepare << "SELECT qset FROM scpquorums "
                                          "WHERE qsethash = :h",
                          soci::into(qset64), soci::use(qSetHashHex));
```

Recommendation

To address this finding, we recommend reviewing the identified variables and assessing if their scopes can be effectively reduced without compromising functionality. By doing so,

you can enhance code quality, readability, and maintainability, while also minimizing the potential for unintended side effects.

ECR - Enhancing Const References

Criticality	Minor / Informative
Location	bucket/BucketList.cpp#752catchup/CatchupWork.cpp#543herder/PendingEnvelopes.cpp#152,215,705herder/TxSetFrame.cpp#793,1046main/ApplicationImpl.cpp#428process/ProcessManagerImpl.cpp#271simulation/Simulation.cpp#362transactions/TransactionFrame.cpp#1307
Status	Unresolved

Description

The codebase is using variables that are initialized but do not afterward without declaring them as const. This enhancement pertains to variables that are not intended to be modified and can be declared as references to 'const' for improved code clarity and reduced risk of inadvertent alterations.

```
for (auto& pending : mPending)
{
    pending->mImpl->cancel(ABORT_ERROR_CODE);
}
...
char* p = static_cast<char*>(std::memchr(buf, '\n', sizeof(buf)));
...
```

Recommendation

By utilizing a 'const' reference in the parameter declaration, the compiler is informed that the specific variable remains constant within the function, preventing unintentional modifications. This practice not only enhances code maintainability but also contributes to safer and more predictable code, particularly in larger codebases and collaborative development environments.

USA - Using STL algorithms

Criticality	Minor / Informative
Location	bucket/BucketList.cpp#733crypto/SecretKey.cpp#122herder/QuorumIntersectionCheckerImpl.cpp#50,719,726herder/TransactionQueue.cpp#1161herder/TxSetFrame.cpp#158,551,693,652herder/TxSetUtils.cpp#54,74history/HistoryArchiveReportWork.cpp#49invariant/SponsorshipCountIsValid.cpp#232,243ledger/LedgerManagerImpl.cpp#1185ledger/LedgerTxnAccountSQL.cpp#612ledger/LedgerTxnContractCodeSQL.cpp#119ledger/LedgerTxnContractDataSQL.cpp#133,137ledger/NetworkConfig.cpp#774,1107main/ApplicationUtils.cpp#816main/CommandLine.cpp#453,682main/Config.cpp#1073process/ProcessManagerImpl.cpp#206scp/BallotProtocol.cpp#1434scp/LocalNode.cpp#75,82,117scp/QuorumSetUtils.cpp#106scp/Slot.cpp#298simulation/Simulation.cpp#153,161transactions/BumpFootprintExpirationOpFrame.cpp#135transactions/CreateClaimableBalanceOpFrame.cpp#276transactions/RestoreFootprintOpFrame.cpp#145transactions/SignatureChecker.cpp#150transactions/TransactionFrame.cpp#412,953,572,659,666,1014transactions/TransactionUtils.cpp#1486util/TxResource.cpp#69util/types.cpp#79
Status	Unresolved

Description

The repository is using row loops instead of the Standard Template Library (STL) algorithms. The absence of STL algorithms in favor of custom loop implementations can introduce challenges in code readability, maintainability, and reliability.

Custom loops can be complex, making it difficult to reason about code behavior and establish post-conditions effectively. They are error-prone, particularly in scenarios where non-obvious conditions come into play, leading to potential vulnerabilities. Additionally, custom loops may inadvertently introduce performance bottlenecks, adversely affecting the application's responsiveness and efficiency. Lastly, they tend to complicate the surrounding code, increasing its complexity and making it less accessible for future developers.

```
for (auto const& b : buckets)
{
    if (b)
    {
        sum += b->getSize();
    }
}
...
bool
isZero(uint256 const& b)
{
    for (auto i : b)
        if (i != 0)
            return false;

    return true;
}
...
```

Recommendation

To address these concerns and uplift code quality, we recommend replacing custom loops with suitable STL algorithms. By embracing STL algorithms, you can streamline your codebase, enhance its readability, and mitigate the risk of introducing subtle bugs. Moreover, this transition ensures that your code adheres to C++ standards, fostering consistency and alignment with industry conventions. In conclusion, embracing STL algorithms is a prudent step toward improving code quality, making your codebase more maintainable, reliable, and efficient.

SHA256 Codebase

SHA256	Filename
0f7c5b47353ed36fbdf60b8620f13157132b9e6b7a9db96cb50cb11d26ca7b51	Makefile.am
b39727f870feb0f54b438abcba8d4c11fff8530b9ccbe5a35db08c3bcc325dd2	bucket/Bucket.cpp
6522c255ffd88ac1deff82137d83e64309b7d59d1823a38be38599aa8c341959	bucket/Bucket.h
e9d019278b3ebdd9017b5060cf6eb8e140a1c6ad636d89c1a46ef9c753ed1b2a	bucket/BucketApplicator.cpp
6fa3c1edfc271164365c0b13c9c049be1cc1868abd8a5f9396d23efa323d4dcd	bucket/BucketApplicator.h
935cb61d04449b8073e2a54da610e614b118c5e25220e249d2e5287f4437db31	bucket/BucketIndex.h
32820aa4cec1fd55afb74161f7b0e93fa553120e27f537008acfb96adec7d649	bucket/BucketIndexImpl.cpp
cae7a0238eb2d63bb7d670fc3561c43b65db62ed14321b7a7c02802b7f22d8f9	bucket/BucketIndexImpl.h
2cb8225d9c222bfcf6e24badced8bde23251812b29f4182d32a739210ab22534	bucket/BucketInputIterator.cpp
97cbf20ee96aabff757bc97b8a599b6df9b7cca4fa7a28e0f2edf42196f6d99d	bucket/BucketInputIterator.h
37ffa0f07059f1cd042d807ab94278e3089d855a51af9a1c607eea628afdf7b5	bucket/BucketList.cpp
197dfb3d524e3a25d0644b77c74a8799f90b49f560b5f2fdade51491eac6f358	bucket/BucketList.h
5637108ed6522264e49d4b2978c71a33f8ab8abc3221a81fe87f822e0821e548	bucket/BucketManager.h
35ad3484b803256223ed9b32d64645eef1691d955004966311cde4e9afb797e1	bucket/BucketManagerImpl.cpp
80861029a5c733e13bba8d5db606d730f93fb606809dbbcddf5266b2689752a3	bucket/BucketManagerImpl.h
4a3d460404fcc43e9dfbd91e7db411aea60d6bb9165a18e1f19e5d07fe975451	bucket/BucketMergeMap.cpp
5aa861a4b8cfa2ac7eb1e7af0e0742223b11657c6fb8a3d0222887294a943ac6	bucket/BucketMergeMap.h
3e8a0b0eff7e7499a9a0d15b62f03f23ef1dfa6bb0dc81d5ec8e6dc505b643f3	bucket/BucketOutputIterator.cpp
4ae54aef4b5ad10510fcd9d78fd6149c077643de8e86e39875d11996fe27f572	bucket/BucketOutputIterator.h
77c4d5ed4443c0efbe2b4ad675bfd5a7791799019a91e028b1b85893a7aa8a5e	bucket/FutureBucket.cpp
a0656b28a6d9d3240c4f95a96d59100747d8e4d618090ab59c99da13929ecd0a	bucket/FutureBucket.h
71349d6a5b26397e78a77cd6113a33388fd90fb441a23882ac4d38aac2659d6d	bucket/LedgerCmp.h
f8f5670227aa48b1372f389289c587933b579f05f2abc5db4e4dddfbc4e49398	bucket/MergeKey.cpp
4f8bccf1adb5a5b4be9a2229471b925da9b3b65daa469cbca290e0e0614598be	bucket/MergeKey.h
698fc009581c86c9b1966fa4017a424528f72d27ed087c9b9f11eb09ac009634	bucket/PublishQueueBuckets.cpp
4a6671e6275d578542fa834a74cf06986cdd1520f2757824a7c45e5f6d3229e3	bucket/PublishQueueBuckets.h
9e3bd7fb42c6d3c8453fa9e37e310c3429b30e52957bddf08a7d8a24f9502fee	bucket/readme.md
9304a868d1cda2be84919d927952b077dbfe38392c0152fb02acc6972b58f622	catchup/ApplyBucketsWork.cpp
6bdb3120d0aab6519c13bf1f5b3c2ac15119b7757fa7b90afffc0677149b51d3	catchup/ApplyBucketsWork.h
c0f7a237767290dbaffb133a4185745941cc5d3cbc8f063985eee2a33c719815	catchup/ApplyBufferedLedgersWork.cpp
af8b8750521d923557196e0fbb7d4bada22b555a9ae900827877116c0ea2ffbe	catchup/ApplyBufferedLedgersWork.h
9c83327d3aa4bf031316a76985b5970fc5d074dd802be45597c8c5f3293ea263	catchup/ApplyCheckpointWork.cpp
acdea1cccce4a51cc88d91a431a4c332d00e4b6f392850717c2b748b366c647b	catchup/ApplyCheckpointWork.h
ad15dbee09f72829565b334f5f39e21147be673ab67691a5625ace1f29035341	catchup/ApplyLedgerWork.cpp
d9370d9386d11ae7bc4c1cc33174b5fb2e738fa6cc065b4d4bc3e119759af7a9	catchup/ApplyLedgerWork.h
4cefc5f53bbd37fefec9432c7d945e527d649a85bf985035e65de83328e3f500	catchup/AssumeStateWork.cpp
c6999604ce134d4786ce699e49be8689164801b4f59aef082c0d891a803add22	catchup/AssumeStateWork.h
3a37524ee18cc2e7c094294ec07fc26f346e543e044bdbc5dd9a907613536e33	catchup/CatchupConfiguration.cpp
2601a26ac273fbb7fb61bcf27697edbf836e4343d42d80b6eaeb1d9a055e1922	catchup/CatchupConfiguration.h

ca8db4c6b47e99fd787aa3f503513194112117631ce1c9590ce8ce04ac826536	catchup/CatchupManager.h
7e4bd2d5e9a5521cee4a0acbc5cee431c596dffcec6259cd1b035c742a2bed37	catchup/CatchupManagerImpl.cpp
c334940d929d8a52fd0b1c606a6c71c6fbd2a64264455aed4352821f52d1ba70	catchup/CatchupManagerImpl.h
57abbd8e4044cedeb82e003e80e91577710497a798377c91220c73cd5adecce9	catchup/CatchupRange.cpp
4985e5ee93d0e2622e7f89c2575124a8b3c1f5483d300e762fe70253bbdc4e75	catchup/CatchupRange.h
a2ab329bfc67bfb33f8606f391f64b074681933db3fb4e5007aab00d74ca02ce	catchup/CatchupWork.cpp
cc8d82171c994fe51fed539fc132039f5b6678c880be824dbab5f92597f201e6	catchup/CatchupWork.h
38d86cefdd9ca77a3b78ac01f5aa7ca122b5c023ae8d550ea540abb40e63eb42	catchup/DownloadApplyTxsWork.cpp
3f060361fb0f0c4268f27c54aefb9c1552022ab3d84de4771ccdad8e170959c1	catchup/DownloadApplyTxsWork.h
eff2a9426bb12d68b10bb2cb5543d8e394562fb1467a88c36dfbdab271ed4355	catchup/IndexBucketsWork.cpp
5fd4ce27ab7ec6a9d64bc3f6c163b2f012218bc45466ade91ad9c1870f03b41f	catchup/IndexBucketsWork.h
e31cd0a6fc7346b3e273a3f0642ccb2c50705528436b1b013eed5091346dc501	catchup/ReplayDebugMetaWork.cpp
258e2338c6acc0d923ccf97fc9a20cf52cd04f7635bd358034cf2684832c93d3	catchup/ReplayDebugMetaWork.h
bda90a8814246f4f7c362e095ffacf9fd477e4020c9b9cf1c8a152de7c5a8c56	catchup/VerifyLedgerChainWork.cpp
8c3170da50861f05727af4ebfa4d5ef35c179f4fa474aa3601fe9fae5037ce96	catchup/VerifyLedgerChainWork.h
503863a8611ac7e3b06f44b681334ebc681f217cfd2472938a742a4978cf85ba	catchup/simulation/HistoryArchiveStream.cpp
4a9a9d64a197fc7ae86cc78aaf51802aa2e41f40e0138f436b50e5da721d29f9	catchup/simulation/HistoryArchiveStream.h
56accd09a2fb0ab091122b88e02edfd224ec1aabad341760770b28f5a258a9a3	catchup/simulation/TxSimApplyTransactionsWork.cpp
cd818038a65fb354e9e6ca8a7996ca08d22a6c01d17d970839ea533a3501515e	catchup/simulation/TxSimApplyTransactionsWork.h
57503909c7d643fdc64b4536738d074f79c4649e332cf1f9a93875c9afc90c72	crypto/BLAKE2.cpp
2a0a52c64963801f47a80fbbec5c749540eb7ac80104144ded813bc95e61dfd6	crypto/BLAKE2.h
5448b7ea84c80e1b8579f1173bb73a1ae49f083d85b7ef9678c8dd440b2283ce	crypto/ByteSlice.h
76d6471726edc39d09900eaf4f232edc4a97a8b9a7828a2d7058791801aa97bc	crypto/CryptoError.h
0cdb6df7779be66bfa6cfec563ca735ae4224849610cf2e310b7c2821b608da9	crypto/Curve25519.cpp
aafea129da289d5ba7ccb0c97f2606dbbf880bc8ce4e667edf1bac2c4b9b8543	crypto/Curve25519.h
ee9b4b907b0d7c938ac61e5e65866aa57d870fa9881b825d72fb32e473e63282	crypto/Hex.cpp
e7696feeb4830fb31d816776e13558ff7cc63ef3c50177cc460a0bb4736cd0da	crypto/Hex.h
473fb6ec4871c1012a763bb9de337b44c0ab65d65e29a3e8cfdf6c8a69b9b358	crypto/KeyUtils.cpp
12250b5edd45e7ee75176a0119a3b57e362cafb20ec4aa5c9a8b09bf433178d1	crypto/KeyUtils.h
fedf34503b653b647d63d72b733f9943c9b7eab9e274dfcd201c56f9b6ba9243	crypto/Random.cpp
456fc9b21ad734cb93ebde8df37377c9c7b0c9212748f19e56bee0a88ba48757	crypto/Random.h
db5f3cd613444feca7b61eb8d46031374141f4e1f4764d9435b2c8ca231e2db6	crypto/SHA.cpp
7affce2ce17fcf80f0e453c58f879a0b6f37d22c51c66de4f1057c82beef9a8f	crypto/SHA.h
72499354e36c4bb7a36d0aaf7863c679b0eb114b453454cc7ea439ea54954bf9	crypto/SecretKey.cpp
2ec0e4a0a098a4accf5921d97ff42b4e32aca2524b8077c1704b20bd50906b8d	crypto/SecretKey.h
bd16317955b518226994309b5fa725dbed8c8ecccafe55ee2cb94733c7de38c4	crypto/ShortHash.cpp
2315917fb61002b4cc5c079e73e49c90c358ea5d1260daa67dc15e8a838c708d	crypto/ShortHash.h
636c1dee7a5ed3c8b4afcce35caf0e060990449795d5cc7424d536f25b659e35	crypto/SignerKey.cpp
155c10c867f0c9b7b47f306c7fe3b708188f2d8e6666cf8899323821b3ac3019	crypto/SignerKey.h
14dde9e734162d1b497a1113fee08beec64bde7611c3c58ee41e0d44ac00b8bd	crypto/SignerKeyUtils.cpp
cc2d4bb36d113af743a6df8366ac3965d076376b7a741fabdfc7a55b852f7a69	crypto/SignerKeyUtils.h
122d4fbcf88cffe3b37e0e5633192d9e3d6b7e24fb3c23a52fd2a193439b7579	crypto/StrKey.cpp
05306aa51dc6d6a78b8e3f689fbf857fe4f7443eab244f39d8d1c56a4c266b74	crypto/StrKey.h
7f45d83f4bd6ed8fac39cc1f89ae44fce3e321783341311fa73b416420c3ef99	crypto/XDRHasher.h

1ba944d10e34f45bbb5561e95ea89553a91913e705644893d1b55b18615362c6	crypto/readme.md
fabe7fb65c21dee5d59739ee81f626f85183ee815f6b0c72bd9fb6daa1265684	database/Database.cpp
5dc2a69c35cfcee0bf1f0267026ccd5b40ab3f7f047839ce96abd9022ef92267	database/Database.h
6fef74cbd286e5a731a1c339b60edc956ecda0b8704005184bca318c7f775ce4	database/DatabaseConnectionString.cpp
90db94995b8f01c0442ac960ecb1e746d16ba7cdde2aa33a587b3c49f7c282c7	database/DatabaseConnectionString.h
1ee4bab6f49fa1e1c93741f8fad4e4d1ee0730bab63aa5bc72ac8d3ca8b7776f	database/DatabaseTypeSpecificOperation.h
c379389d73190e5519a3f6db60ad6f76f65503d0cee8c53dc987fff9c9435752	database/DatabaseUtils.cpp
18bdaaa319f381ee1460244154f3e326cab21eb0580eb7cf220ec219a8ed7e8e	database/DatabaseUtils.h
d3fe3451c513d782ce9c4346ebfd312aa2b4d99085f39730f69e619e2117b2aa	database/readme.md
26eb60147fa4b2e0fcc3609413a0d94e28185f2e7221688b6beff716688d792e	herder/Herder.cpp
fc5436bf80f26bebd4d67b0bf20b840bd2906e504d6a452723adf92e03014e9b	herder/Herder.h
c447f1e614788ec7f90f4823e76d840278751f67308d7052a2966eeba2084c47	herder/HerderImpl.cpp
cad1dfef5c10094ed18611a187fe7170b8ce9239105fd70ae8d9fd294c95162b	herder/HerderImpl.h
c16ff6ce3f6ae5454210d536f5032cf2bcecf8cbf56eac5b3451870ba56404ee	herder/HerderPersistence.h
482f8fa99ed83ebc39aac7a0209b930fc0cb20f28ccfb4b618305856c68c4df4	herder/HerderPersistenceImpl.cpp
6bb21e8abec026cbc32368c1d348e82603c64259926af3262b6eaff9293c1e06	herder/HerderPersistenceImpl.h
c581acfb91fd43d6dad35eca4bf38bb6e31f033d6f6dd1d2a6df7c6c66b659e5	herder/HerderSCPDriver.cpp
6200247bdaf249f24603c9e55a69d221a046b20bb3a88caf0b970cab1ddaece3	herder/HerderSCPDriver.h
e46da349be642bbdd330847b30e58bd9d80912f7224149c59b7138600b87280c	herder/HerderUtils.cpp
c891dfc92d6d8581bb0cb11c3bf3e89ecf5f2ecebf4b96a26bb47cc0ea546794	herder/HerderUtils.h
c7677c967b9efc2ac65e2e20d05d87825ed5229d28020839a81a6507d8d1aa18	herder/LedgerCloseData.cpp
d62d3b97d03a1a177e4bc869cdebeb78605f83670b000f86868b08a99b384c82	herder/LedgerCloseData.h
48d8739abfe6005b25be99ae9c746b3a760fb087d2ceefec431d2b2f14461fee	herder/PendingEnvelopes.cpp
b8ea54af3f2bbeeeaace0f7a08114773de08d78532dc3459732ca18af67716a7	herder/PendingEnvelopes.h
8eddd3472a1cd75e30c3590e1bac61cbe21c26b297bd59d190aba0df5f5fe69c	herder/QuorumIntersectionChecker.h
a0086e372e70508f8923d8016fa354cfc9bf1cbf631b6ec21c6a541337134a7d	herder/QuorumIntersectionCheckerImpl.cpp
a4a0e87a9c1a99ca544ac84b41c55521cddb5c232789d2bf8d54364f70f44887	herder/QuorumIntersectionCheckerImpl.h
6cd75df115cb031cf023a64f32530edbb53879b9adef54e17b00b5ec9e5cf20b	herder/QuorumTracker.cpp
551f0fe2cd00c1ed85e8bacd8ddb70f40ef8b1380ba3a1140666f075ae4ca379	herder/QuorumTracker.h
ada12686fda3a7d6017da2113b2ab56235131dacd9917f35f6c92ea6d9cbcdc8	herder/SurgePricingUtils.cpp
cd54f16c3b4c538d955682f8eb250255345530e481b9cf7eb8fa9d4559b2969c	herder/SurgePricingUtils.h
05eca8f94a16cc72dcc5fae9eebae95d933db2acd50ebe22c47e7845a63928fc	herder/TransactionQueue.cpp
aa190970fb1a1adbbe09a25b969cccb4693e6128e40fad8267e9f4d954c41f7f	herder/TransactionQueue.h
523132cc06bee7e85106368d04bd8a7ed23747113454b69beb8cfcf1f949f1a1	herder/TxQueueLimiter.cpp
80a6c28730c2230c49cf87f2ba5f8bad71cc3d766f077ab4bea9aff7b9129c6b	herder/TxQueueLimiter.h
ad83f1c1279139a5277b096ca956c18cb637f1928a5e0dd4927434fc9db4f4ec	herder/TxSetFrame.cpp
e400ccf68bb7424762753332e7c35c476fbded7109eacbb3b2881159b2fd1ed83	herder/TxSetFrame.h
2ad649676914f5fe7fc0890b6e1cf6c988f20275ea52baf0c0390b079a94b49e	herder/TxSetUtils.cpp
dea719fd27db060c346604f178cf4d201eb3c52c3e4dfd864d7f2736d97d8f0b	herder/TxSetUtils.h
156feffe6907d5008840151c4509113e791f6cd7179e91e25907465136dd2e52	herder/Upgrades.cpp
7ef65e86fc4a43c06d710e07bcccb1ceff9cc31ce189b83f4180e72f9a0eb38d	herder/Upgrades.h
df521ce19fbcb2fecec18c0746044eae3e37dc7408a3d693ea1ca003489ca5de	herder/readme.md
971073bb270f5e585f4451bfcceb84b4c8116f0651e5e4fae9bae07ef9d0a4e3	herder/simulation/TxSimTxSetFrame.cpp
dc0c94250310c59fd9008edca067fe7fc1b170b52447d0241568a8c4111adbe0	herder/simulation/TxSimTxSetFrame.h

d64eb4b4b466a6e2a4779fd408d383875d80a5fcc65dbde717a8f8f5cd117e3b	history/FileTransferInfo.cpp
1fd8877c09861a6f20a3b8abe172d41297da3512151325cacecce49f61db0834	history/FileTransferInfo.h
9dec603543da3dc1b79e73f024e18a6496c32e925745f7346c4ad6cb18b1a624	history/HistoryArchive.cpp
c92e3f96c58a6247e3d73783156139d35cf4d3a790dcf985cec58e2a0b00e063	history/HistoryArchive.h
bee830e43623820b68dc00a9e9c265c244e149e6aca996cd3238ad4318e62e49	history/HistoryArchiveManager.cpp
85b2e3d9829fbc6fcf662a43020408c7e97b981d52804f39169a06e8af446fd7	history/HistoryArchiveManager.h
337fcac3c87b07ff05fee2453a0e87dae8db1bbd3fbb69ec742fa09a3a0efaf7	history/HistoryArchiveReportWork.cpp
497accda6d955d46c19a0019f765fed8a4f919e26930b5414fdc117a668ffe5a	history/HistoryArchiveReportWork.h
e4252fd1d21745e37065c88063e74203f1d4888da97e66901bd2be3f3a0117d8	history/HistoryManager.h
4c27107373695fb89da6a5ceef934e322a698d7e0575dd6f606eaab257659deb	history/HistoryManagerImpl.cpp
75631ac055bd533ff3578fcf614f8ee510d71205675e925d9aea7b2e3731648e	history/HistoryManagerImpl.h
43dad40da39e207375a0dc08fe008729d1ee134074ca7b081a7fc95ab6d4f653	history/StateSnapshot.cpp
f069d8f742a9bf4b68a1a27ffc2a80b525f11c057b1a534dfa9958a8efef98bc	history/StateSnapshot.h
bfca1106c6b60bee0cf17b6bcc0429194401ccbe336d01e67a1397c0c0d236c5	history/readme.md
11796e63b35fc479776ad06bdef48d079f00fce1031f3230d6bce8a304a032c2	historywork/BatchDownloadWork.cpp
7c5ed667c53609123897e28f093eb3fca7dab00db00b00b243c9619aa959bd23	historywork/BatchDownloadWork.h
a0d810abe938a6b37ab52d52454c545ec7367fdcd361f36c33f26d795d09cad4	historywork/CheckSingleLedgerHeaderWork.cpp
d47f07aebe2e3b3c5cf084f6d76bbb7842cb421ee80f82dbc34abb388ae87a5e	historywork/CheckSingleLedgerHeaderWork.h
d78357df57916956c727333703d1326e512ff51e7fed990b9d58d2e9b2885ab4	historywork/DownloadBucketsWork.cpp
01d9bd97e063d11ec3532e5db78d9dc2988d10d76212dbb10d6b7806622fd214	historywork/DownloadBucketsWork.h
6cdc6b2031ccb68d5a63bfc7091e8bfaa48d3b5b7bc84e176325f94cfb3be5a2	historywork/DownloadVerifyTxResultsWork.cpp
4537b4fd064b4283fa50d099f0488874eac0926ca35d4ede11037dee011be705	historywork/DownloadVerifyTxResultsWork.h
c772a2b9c1e5b63d4d1b24e8e90dbc41b62ba43b142cb3a03694cddc429ca7be	historywork/FetchRecentQsetsWork.cpp
38d58b2869167f67432c39a9134a07876b3a2ae777dc420210d914486f527407	historywork/FetchRecentQsetsWork.h
8212d5144ec2dbbab4ca570e25618e9a21fee8cc82f9873a42b26ac73f5b6e55	historywork/GetAndUnzipRemoteFileWork.cpp
37b80d78337d90fad1dd442864b86fc8f675d0a38c0f43dee1df0deee0cd6ef0	historywork/GetAndUnzipRemoteFileWork.h
1530888ff8aab2124edc805c08419fd89fd5129cd286820e65a27166d0e78b98	historywork/GetHistoryArchiveStateWork.cpp
b2461803f8273514f08eb30b119e9bffb09056d84cb10f05ec4f9d6206cd9904	historywork/GetHistoryArchiveStateWork.h
f47d70a15d4369739f2d77445a088bf6d6746e070d52e4d07d448d2af85b1faa	historywork/GetRemoteFileWork.cpp
0bbadd9984989b8591a2c15d30227c86f8babb086aba7a722add62a5b868d502	historywork/GetRemoteFileWork.h
b421aa373d08fa47cbebb2a420ed2ff418bf341956510f5a1aeeb6b32fdcb3db	historywork/GunzipFileWork.cpp
dce2c07310009d416e2a9fa33b0549cd7b4eb018044d3d142d062a18aa4f4294	historywork/GunzipFileWork.h
be77161e0888d7725ae77b3bd1a7a023aeb80b59e5428585abe9f50c5f484829	historywork/GzipFileWork.cpp
329d6ba08d6d25e9d0046790e44c75d8bf1ba8c8b0c7d22c293eea544e0e9287	historywork/GzipFileWork.h
b7871133024bab67b02fa6629f9ccae40ebe993900f3d12f80908c71a59d73f6	historywork/MakeRemoteDirWork.cpp
7f702851f05add9067918267d186f124a7c43faf420cc1c0c414360a523c213d	historywork/MakeRemoteDirWork.h
116544bc07ef459d5a4ca4215fdfc6240b15af8d6e5f7d2a482928491513e293	historywork/Progress.cpp
394230e839422d090a9fc97a4753064ab82e7a960b01831c8996bf2c447eddc3	historywork/Progress.h
6d9ebf6c66f90c9ea147974a38c25baeed0732cbb031ea9d97fa685a7bcac2aa	historywork/PublishWork.cpp
150635eb3f22e4f0c8062b7a207f3fdaa810fcfe9d7916f03440876280f93cf5	historywork/PublishWork.h
ddf0ea1cfa0433109ed66950661c4547d9700a133a78c8d4ba26394eae549688	historywork/PutFilesWork.cpp
52cc58738a7cae85e2d64c07c0adaee61a1a53cbc43c234b5073f64f6c6ebe07	historywork/PutFilesWork.h
bd0ce34833dfc0d99b40df79e472a8542f89b065eb11ad867f9722cdf402c41a	historywork/PutHistoryArchiveStateWork.cpp
d210b76dbb0767792110bfeefb3ec5cc71c935b3125ce8590753e44dde5d506e	historywork/PutHistoryArchiveStateWork.h

7876b6d0c3e4de27bdd90a54b7d3fa507e8e93bcdb9529f201300196f0891fb3	historywork/PutRemoteFileWork.cpp
1da865431ea96729101ac56c9c0d465fee8ece4641fc1b85e4a18dd0e3863dc9	historywork/PutRemoteFileWork.h
189eb934009962bf4f41ae8d75e11044fb84fe7af59e123733e8c3ee02ec182e	historywork/PutSnapshotFilesWork.cpp
6e1f7599103b773581a067c150ee24fda3f675691478236e3abb07123ca4e0c4	historywork/PutSnapshotFilesWork.h
6b3f8fefa2343b7fb6ee702352ed2127d997a6643da43dcf142a9709703bd078	historywork/ResolveSnapshotWork.cpp
6abf697f3a23ffd47061322f71b69d4e554473dfb8944fce8769cbed5408b39d	historywork/ResolveSnapshotWork.h
eacb68b9f2e391e38725c415db068cc7fa09f6001a0a4a6c5c1d41ca3a622f13	historywork/RunCommandWork.cpp
34c297310d670fa2691aa8c1a3105d496a4b7833a996b21df2c871fe48ecd797	historywork/RunCommandWork.h
fb65e57fb2cb5143e87b7132e2b3903cd2dc194cd6a93ce30c2796f5cd3994ed	historywork/VerifyBucketWork.cpp
3574ba8de6ad24a35dab630b58c2e16a7c6640ad0c588b80fa0c8b604435809e	historywork/VerifyBucketWork.h
e1d344d0190e2325885d350cae067fc214dffa83d9000fb23dd093b02b2fd3af	historywork/VerifyTxResultsWork.cpp
719a7748897e4fc77894b2d606e5a1d7e8329edd3bf324c0d8d7b84c96e8691e	historywork/VerifyTxResultsWork.h
8b47454909fdb179f825ba30bd046d784f40581d12a2d2542d3a2ebe0cfed622	historywork/WriteSnapshotWork.cpp
382c433eda0528321878661678f6ceae919db88850572acff7a29549a19b18f8	historywork/WriteSnapshotWork.h
c625aa62e846a3887ab4b3faaca6a7a48fa839486b36713b97a1358883e9143e	historywork/WriteVerifiedCheckpointHashesWork.cpp
da59e46f863808cbcb671860e07529f1ae3ef040356940b18fb0672cfc9da527	historywork/WriteVerifiedCheckpointHashesWork.h
a4ad4de5c3dc353d79e107912caaab3a460cb3cd45e60205bdf686d37e3ada30	invariant/AccountSubEntriesCountIsValid.cpp
fcc73720a0f9252c8c01827dd041eba91ca9fc4ef407ce730dde3afbee0d6142	invariant/AccountSubEntriesCountIsValid.h
35c42036c481fc5774b03010c72d2f73db1977a2cbc64357bf93c17385a4af78	invariant/BucketListIsConsistentWithDatabase.cpp
af3c27474df59efdd9b6eba71816c34fa9f2c4cf429fafda879a12749898fc15	invariant/BucketListIsConsistentWithDatabase.h
75c907b7183db46393ee965da62a30c3e72a315fa3767c63f774861d650fa283	invariant/ConservationOfLumens.cpp
b91729916784cb69c8bf4955e0223ecf9431d67582aafec0e692522bc431193b	invariant/ConservationOfLumens.h
1e9649cf5cb5e4547240af7f2ad8d7ee3538b524d81e07ce4a5adb49f81b4e4c	invariant/ConstantProductInvariant.cpp
ba4c486dd2438666db3557da719501f8223a6e33c70109ae23db0998ac1958a7	invariant/ConstantProductInvariant.h
29b5f168321e51a77f602ea4f93a5298be5ec857293fa684afc11fd71db167f9	invariant/Invariant.h
5fdb6e941b2562f68838113c23763818ab726b1287e20a1aa4dff96d54b196a7	invariant/InvariantDoesNotHold.h
4df4589b12df3e27b2c5f0264f0656da42da210e394e15a5996d4130e2e83e5f	invariant/InvariantManager.h
53668302b3ae40a81554bc334eb4384740d0b1c9187739123751acecfd7c4fe3	invariant/InvariantManagerImpl.cpp
1d96d5f62404d0a3a624281acdc85963c40cdba24585c8ba56daf1535611f729	invariant/InvariantManagerImpl.h
3040b65ed1b8617ef05eef1e3847cb8395985fe9aa3c9f63ff432ab9f3786855	invariant/LedgerEntryIsValid.cpp
ac6d283b25023142ccbb4a886066a8ee31cbd9e01d4fae149ab28e9642fa2c48	invariant/LedgerEntryIsValid.h
31117987e0528a553f7e19d984447e3e06bc9eb9be0864837f94b0a6dba0b2f4	invariant/LiabilitiesMatchOffers.cpp
14ed7127d287a6c5c086c1aa16b6cd67a74a2751c8f09607c89561a00983ed93	invariant/LiabilitiesMatchOffers.h
8d2a00ce0c40a03b3d6bd6d39a3196fd58df9808cbc83a27b934ac7176c6aca7	invariant/OrderBookIsNotCrossed.cpp
27877f1495648d7b090c1731a7d5164aefbf892512f8e4ef713b49cf2b2ff4d0	invariant/OrderBookIsNotCrossed.h
217a21cb606773965cfc51344eb7988e0e8c93a505af0d2744374fa211d3b2fd	invariant/SponsorshipCountIsValid.cpp
25a9d3cd9e3cd964a8c5e509570f9422df85a91f6999dddaf97bf78dd5796518	invariant/SponsorshipCountIsValid.h
2b7ce551183a285aa8519930188f93582dc44550ab064c93849fd61c01865042	ledger/CheckpointRange.cpp
3cc77130496c3aef6b5305fd03b8e0e0e3e9e815c643068273a0561579471a68	ledger/CheckpointRange.h
65c54307d45f301c65f5c283544f2d9dedede9cb8273f8d4b28d95af22f2f71e	ledger/FlushAndRotateMetaDebugWork.cpp
3d926c0cdfa0d2299df05c14fe184367ecb2afa21f433d2f5d3c5fa422afc83d	ledger/FlushAndRotateMetaDebugWork.h
3eeb424fdefd44220c84f1760648dce992cdc6a5ca024d293147534d2341ef39	ledger/InMemoryLedgerTxn.cpp
98b037cf0c8f5cec360d404e2999dceb887fe20c6ef9bc8bd464fda2c1eab2f8	ledger/InMemoryLedgerTxn.h
4749dabb4a308dcc95b155216e2168333eff2df70dd0e7060b50e5d903146e61	ledger/InMemoryLedgerTxnRoot.cpp

31f66d7750d6e1643a55c3e9d6b01bff4a27a561e4d0ea20af9fb5504bc16424	ledger/InMemoryLedgerTxnRoot.h
52db07ee9e7111079cb1cb7c709363cb58c9dc73758a39d10a03c5312193e40d	ledger/InternalLedgerEntry.cpp
841cbcc15de1297e5e337bf3b072354ea7e0266df798353237d5d10eb47f3108	ledger/InternalLedgerEntry.h
b7eff2ef4f2bee16fd9f6ab93b874c1cc2be9cfd143dd5cc63b73ecceec41bd5	ledger/LedgerCloseMetaFrame.cpp
2df5bd3b1287a1aebccb134359ee86a54215e9e2a8cc1c6f97f9d22c1d7bee1b	ledger/LedgerCloseMetaFrame.h
9de614fc8fa3d3968e99a513f73f79154b4927dd228677771f7de6005e4e0e30	ledger/LedgerHashUtils.h
33701a3e1f31f748286388ea36817db768819a7ada4ce27964e67c7e45fdb105	ledger/LedgerHeaderUtils.cpp
f51a9cbb7ba3c93f7f011221f716af1fd7d70cde2833c38fb2136d0315bf2409	ledger/LedgerHeaderUtils.h
f8b38afff87541105e46a26b512ffcefeb44df980dc9a26b53ca6ffce719449c	ledger/LedgerManager.h
cfd692aaab65409cfffda6cc94dd1351b1b101dc6ee89a7cceba831c7fa4c4d9	ledger/LedgerManagerImpl.cpp
f643be2faaa8af39336a45342591b98ae1df0e734a3fe5d1fdc1689ffe001462	ledger/LedgerManagerImpl.h
152d38c6791e072d8aa806febae6d70141a8f3d83564ec472b5bacf73e0d2a6c	ledger/LedgerRange.cpp
8aca6669e4da5b6fd6d325605525c9466bb54b4416feedea32c8ddf99200dc32	ledger/LedgerRange.h
7014c01c9f50eef37515a4ec6a8b86e98084991dffbc3a375d8729b52bd7fdf6	ledger/LedgerTxn.cpp
72009959da139f68b6b7b84cca66d6ca01ff648477e72a835c36ad9de74ca667	ledger/LedgerTxn.h
5399563d9bd53213a82a3a4cff5a5a030e6b419ada868c9463dd5561b17d5f1f	ledger/LedgerTxnAccountSQL.cpp
0d1e7e4ae35eed45d07e8b2a2b044db74e2445a6131fc46b0939c4dee15ff5a7	ledger/LedgerTxnClaimableBalanceSQL.cpp
9723fe5bf6810165dab7429efc78cd13d2cad5cb439e61abd04e592d46c02307	ledger/LedgerTxnConfigSettingSQL.cpp
f5e1147b6f9d6e8c90bff8fabfe724af1e4daee7d4e8d0b3ad7546ef828539bc	ledger/LedgerTxnContractCodeSQL.cpp
dff0c09bca2f25bd99409fd583be160d309cba5e9ddc4b18359a0ad4fb81dc40	ledger/LedgerTxnContractDataSQL.cpp
c0a41243bb0524cb5b99be074443ebd17aa7a586e024dcfd6fd8e2913e44b668	ledger/LedgerTxnDataSQL.cpp
e30412c0913fb39dd0a76eb451a4bbde7d1887a2cc0ee9f6a1d85d29e7a68a74	ledger/LedgerTxnEntry.cpp
e1e0f4c19a14674c710bfa8d7407b9ef67dc47504f752737fb057f352bc8d3d5	ledger/LedgerTxnEntry.h
cc74ca43a8be3d4a0508b7bb62ba0edd849070d60b869a3ffb4492a5ca28e2f4	ledger/LedgerTxnExpirationSQL.cpp
a13eda51d9ac68f061f1e20a8a347a0fb035a0a6698e633cdfad7aded193e498	ledger/LedgerTxnHeader.cpp
8e8c183bc1f946b3ada083c267828787b9cff47fbce1638b80fc5d164961f337	ledger/LedgerTxnHeader.h
062f01d77e4c1b764440198bfdbec19cd04d5ee5dd4f713d80b6ba9e3a3cb86e	ledger/LedgerTxnImpl.h
8b0fe4b16e254c86a403e7861722de8d204ead427251737155816171f835bcae	ledger/LedgerTxnLiquidityPoolSQL.cpp
6b89b2d395dd2c6112e00aabb0c1c45dca363157c072c62232a5ed8fc0412855	ledger/LedgerTxnOfferSQL.cpp
ab76525ef2236b0b1d4a41f009db16859003df772e245c67d7b9f3bd857f15c1	ledger/LedgerTxnTrustLineSQL.cpp
e08ef42120cd2876b7ae300ad982418e32070eab2bcba6ff02362ee4146fad81	ledger/LedgerTypeUtils.cpp
c3667e8fd8500ee92950307d5310b81238c969e9b28a54da850baf0b31e16ddb	ledger/LedgerTypeUtils.h
ff987c184a556eb459b005b50623eca4fbfd18695e2302799de34d343cd72104	ledger/NetworkConfig.cpp
55f2d96b22d129a73760be541dc95593af5b4fa1d8f6b5c1a58f2a7369d02722	ledger/NetworkConfig.h
251ca6a7ad4d6e6c189403c02b1b7d855df6fa3fb55753c278de2eb0a3dbe4d8	ledger/NonSociRelatedException.h
3a9401b0d67212e05e51e62076ef3c8fe68fc1735ceb72ecd1286bce53fdb172	ledger/TrustLineWrapper.cpp
b800a0975b251544f27d9b569626d8293dcd5e14eae504667f3f0fc429e96d7c	ledger/TrustLineWrapper.h
7b3b747f103b2f5c4b9b7ad95586cb96a0bc604befdc3c835dc25c80a9e1f328	ledger/readme.md
dd0b3d890c2db5493d944261c6a5208bf7b9aa98f9e2ceafab3be6b834f49659	main/Application.cpp
504f1d9d040b128505074dad8840b9fef775435500481c223f22cae77180ad05	main/Application.h
c8233ba3c0469cb7230e1a5d1a6686809c73e2156107b3de224cc172eddfb016	main/ApplicationImpl.cpp
e31629e9c75f63647f3bdf15bcc1f3b5aff2a7c7c9afec4446e6e03b8393e405	main/ApplicationImpl.h
cee840c5fa42af1b9558d80f017a0bd2881e608c4dc7770745cb80356b96248f	main/ApplicationUtils.cpp
2fee2fd2ea88e1c1171e01b8052b9d93e2d0005f481983d1eb909decc6331a79	main/ApplicationUtils.h

b62770693ea01cab0648f7e7050d826db34b331f3283b41b1011c34f410c41a6	main/CommandHandler.cpp
8031092b4215a1fc2fd3faa50ad54c6f7138d1a88d14351521d0cc246909f621	main/CommandHandler.h
5f7d085325483d6cd921b73f7e8ab57d652a4aca5a0228ac9730d8eeffc000f2	main/CommandLine.cpp
ab2fb513827dc201e1ceeab2a07fd92d9cbc8e2d2fd9b60037bb38ef234aed48	main/CommandLine.h
8d999468506d7581574fbc3f95b843e614e8c25b7d200a2dda6dd3829191036b	main/Config.cpp
08fc9635a54bd45d9d71b8218fb85ceb3c360ab8055a0142ffced39a6445bf1e	main/Config.h
b4f27b91b40a0d872a43dc01962bb3459930a4efd9b9a85c3bb7d63d5528788f	main/Diagnostics.cpp
7f58ee8c9efa9ee5867f376829a8c26c54e8bfed8a9d427d9d0a3a84fc8fff0e	main/Diagnostics.h
1aade4c2785cd8078c30a7244c549529b8b3f1e8a1688e3904d505d4bd7923a8	main/ErrorMessages.h
80bf534c8529716b2c5b0d67c483ccdc614c72b51397435551c26b26d3ff0d17	main/ExternalQueue.cpp
e7c2ef7cd0140c739a83b05c4b19cbe6beaa8ee37c03dc020b829c2b250d5b9d	main/ExternalQueue.h
aaa2c900b8688defe30c92bf8e0966575348a4cc8aaf336346ffc49e9996973e	main/Maintainer.cpp
5fb98750041102dbe6431ce4b7e66e7502df59169aa1b6f65be8655033bbc453	main/Maintainer.h
2dfbdad0073b1141a83fc873f4e347f01518e4970b59ed14251e60486f8655a7	main/PersistentState.cpp
5412ddb377ebd1cdc9870713aedd42c38529ee6c41b3f98f2fa6d5e7e8702853	main/PersistentState.h
0dd5f853480020efd4cb95831e39b9fdac9b6b117d969fbe07ad20c47001595a	main/StellarCoreVersion.cpp.in
062c7d56b68531985d58b97846b7ac1e3f9516fcb60349164b6dbf65357e9976	main/StellarCoreVersion.h
d6a3463d257fe82f7c5b07c6b4096e33fdb1dd7f973bbfb62e3955372aee36e3	main/dumpxdr.cpp
acda40a2b9b0dc7fd018ffceab7aec3842b5ca4093a1a31f764e3a4b4df8a0c3	main/dumpxdr.h
b02bf7a7179acdb2082a29c58586846da692d7407a7d9d4d7f950012d273eb59	main/main.cpp
f443a8ec990cbfb7279849297ccd954e6e49ede5c16eaeffce2438a6114d97b5	overlay/BanManager.h
d72cc29259417696e0a0dc21d7902371679849edc2d32795abdc9f8845af160c	overlay/BanManagerImpl.cpp
92b65199ab68e28c8d686b56afb21232416e4e7911430a5d1fb35c691ec04c51	overlay/BanManagerImpl.h
df67ce8e567db8f58565325bf40de8593dc90f5ffc5864ec6be3ed542b6022d5	overlay/Floodgate.cpp
2e6d47a965acecc149e93f522a48d1ba69fde47665f4d6be3f91c37a6d44f175	overlay/Floodgate.h
20ed40fca0392356f99b819ba61c8fa14e4a679f52aeb09715ac819cdb668d6b	overlay/FlowControl.cpp
a5b2335756aa986e113d8477d52cda27c11d5a0e1ad45ed7fea4bb186a282a0f	overlay/FlowControl.h
7e9d54cb26ee589bc9b6d49d4ac1c986f17c10b3e6c9148830d4dc4aabc2301a	overlay/FlowControlCapacity.cpp
d10ba793448581d42f23ea6a56a4383ceaa580e540bd5dcf16ab69f7f2e61c4a	overlay/FlowControlCapacity.h
4039353ece8102413452d9dfdc7f0f5855f67359e46f4e8aa60c90eabf81461e	overlay/ItemFetcher.cpp
bfb9703528f1347ad551ceff4bc71971d2599ac02d1e4b3d762338eb996da67e	overlay/ItemFetcher.h
34f0d7f2d65575876e831293ce5480b420cfafc452bafc08a11e17f8547df254	overlay/OverlayManager.h
055c727f49c63b1d4b07f4d1e06b2da2c5ae75f899da4cea285fe5a05bb6a13f	overlay/OverlayManagerImpl.cpp
de76415b58f82c8139a6c792d67dee076f3169f9a1556d6304802f56d9769c35	overlay/OverlayManagerImpl.h
69a866d4474cfb059064a48efe8e64a17507dab07eeca0ef3bfb76c8c2b8f76a	overlay/OverlayMetrics.cpp
94e9bfc829e8dc00588491f48219f7a97fecdb51074c29a5aa83d2254e077d36	overlay/OverlayMetrics.h
c57d1e5c81fd8e971714b4107734fe50a5574315bf5ea7c06f68d5b47db33afb	overlay/Peer.cpp
dc3b3380c0c71bb3369986e0387aff92c016ded0e15ed4d947f163c089789257	overlay/Peer.h
3d9de50befdb0769e462638188b761d397f19fbd4f51cfebb27a57ebdfb7209d	overlay/PeerAuth.cpp
5cfc897d73e1277cbd22ea3fa77ed8a2dde95679a7acf48dc29ba4bbf860f0d8	overlay/PeerAuth.h
7184dcfa5bc9b784381f9cc5b20f33204dcf176f331e967222dd6b1f06dd2317	overlay/PeerBareAddress.cpp
9fd4e3fde5e1365c2df37c54b2f471d765e616d83071fb00f5fdfcafd05fb8b6	overlay/PeerBareAddress.h
3cbb72a7c1cd0375720e138caacf84ceb5f97432319acada514edfed16c4eedb	overlay/PeerDoor.cpp
4a1b316bd019d8a710df7a451d06205e1aa575f29cde679974c8e3b6b71e4e06	overlay/PeerDoor.h

bc8ea1813c6374fc09274daccd9185303f446e1fe8b0dda88ccda7a983cad05f	overlay/PeerManager.cpp
f1378cf556a1e9fe3568897c12015c92868642f5a28a25dc431c4efaa4a72b7a	overlay/PeerManager.h
c410662dc7b06b40ea4b2d8bcea1dceb742fd742ad8730224898438c04a6444c	overlay/PeerSharedKeyId.cpp
c3f2d6389bb953cc58ea449dab7746abccb2576a223bcdf0cef7da9da4423cad	overlay/PeerSharedKeyId.h
29befe791a21e8c91036a4c5358132df72ddd9d242b70cd8e9f3f0858defbb5b	overlay/RandomPeerSource.cpp
ecc562a5e8ef13cbf946be1c7ff20bc6db0f852d857e72376f6bd05bb22925f9	overlay/RandomPeerSource.h
d9b19571254f86f3b5e8df98dfe4aa8323ee6ff9896f4b18a889c444d23d905b	overlay/StellarXDR.h
83602360c96a5095ba336f8ff0b69c87b721d0c768bc82bccf7b4557080d02b9	overlay/SurveyManager.cpp
38a6f27201ec22f816eb8ccbf77df4ca4938059ed20326ce64b928ba1ce3138e	overlay/SurveyManager.h
9d60ee1771df76d55e98ee514bf2603aea1dfef10d8c9e05adff500146e22416	overlay/SurveyMessageLimiter.cpp
15460aefadf357ddae29a7fb6ef03154c84470bf128dffc24bf1e2faa3cef781	overlay/SurveyMessageLimiter.h
88405d844871ce8f97e7e192932cb836ce84eeea86ec0a641b9b9424ac3e589b	overlay/TCPPeer.cpp
ce4f4da1073d891f67db77a45ced0144cb6c7238723cd252aa82fa6a0fbde488	overlay/TCPPeer.h
64b3c8913deeb869842da47eafe86589fd3d95b7c4d89020624aad944bc8abca	overlay/Tracker.cpp
2b542f02ccedbab5c0b1ccca45cdc3beb387dfb153af8987dc8f530b65df5d54	overlay/Tracker.h
cab2b05554b75a82ea4885a611caf814024c7641569239dc21f9125bfd0bb43a	overlay/TxAdvertQueue.cpp
4933684f620f336e579c68745141a3e45c74fc898a62793372d5c31b6b29d4da	overlay/TxAdvertQueue.h
797ca465aed8bcd5683e2ce79b98285f7f27a4c95f71fab98b56352b5dc14279	overlay/readme.md
6f4fcb2fde22f75e7fd8643221322e85dc7a9b384605c2c60bcfcf51ed392559	process/PosixSpawnFileActions.cpp
2ad94df0efeeaa7bdd26788d6bcf6f0954cae9a3c44bad243c6fa13be51c64d1	process/PosixSpawnFileActions.h
f495c4df0c333412a3a2a3bb60f79bbfb223520ce1356ebe1c561ae8847cd1fc	process/ProcessManager.h
754bd2be62cfd09a9f5c22a34b286ec43d4d962bb0664fefbb85795b8ffdbcb0	process/ProcessManagerImpl.cpp
42f236959f75881988b49b5043b3d307f20b4d7484daebe7b92599f605bdd40b	process/ProcessManagerImpl.h
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	rust/Cargo.lock
a1cc1e139c88781c25cfe290bdcf75c5448306474d3a2c8125b0e8a2f0a9535d	rust/Cargo.toml
80ac562b5f5355956c36a414d3b007c180eed1c5d10467718fb9cc42da88a923	rust/CppShims.h
81d4c5cf0915efa18ec4b45b002f206aafc55cd555a1a0c66b1be3087a10dc93	rust/RustVecXdrMarshal.h
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	rust/install-cxxbridge-cmd
986e73a8b221e4019faa66f226207ca1938eaf7d4d6a95b12f17d7320fb337a6	rust/src/b64.rs
c4b8c9d1d3b99536a6d4062acff9bde0005b8dfef962ca53d6ceedcd174fcde4	rust/src/contract.rs
a521496c3e49fbae3d75e46894da202c68e345d4adb7dbfd5f54433a70686570	rust/src/host-dep-tree-curr.txt
51910513b44d0fc6994ac117651199c50ee10e5f0a6a26c7d44118e2483eca4a	rust/src/host-dep-tree-prev.txt
9aa8686af25160b1a277802d3108d4da9c8173fdc3379dbbaaac24d4d71e9c05	rust/src/lib.rs
63609257bc0ad7bbe643bbb61afd0da23ca8f6a7702745c2b90601567ce06f17	rust/src/log.rs
6580efc73fd35d7764936e4aa07c629a3992556003328beb35cf472d737ceb77	scp/BallotProtocol.cpp
3f5781439b29f9f85dca847472b8b7da869ed7082253946b60be18e88b898521	scp/BallotProtocol.h
59ff03b6dc837de4428111f42905cf91c5b2e6ef34b92e7e5ea18b6bbb792c79	scp/LocalNode.cpp
8cd72ef5ef4d0cbbe7f6f0c09e3ff481bd37383c3cbec014778e8272523e878b	scp/LocalNode.h
edb10e2eea61f53235282ac06055127db13429f1e776ef278c0fc5cc72bd50e3	scp/NominationProtocol.cpp
ac8435662f13f9a387904a1a4987b778442a879a88327b93a0aad5282119708e	scp/NominationProtocol.h
ef19ed2b8dc78c720cdd85e7628988614e912cc80573d6cde5e5ee390662b13a	scp/QuorumSetUtils.cpp
8586adf557f5f4c29b3089957aacda781dc728ea3cf3c55495d15c11f4b65546	scp/QuorumSetUtils.h
596e9bdfd545c5d8d11280aaa7bfdaa8f87039cc3fa7532aa343786214e58707	scp/SCP.cpp
df3da991a506f206f1b9ff549f07809c43088b3f58c39a691bc17dfb14d9c27d	scp/SCP.h

7badf4799ff8b01a07beb0fbf74a212ac53437f7075913ee5415485b6c83321a	scp/SCPDriver.cpp
ee4a59d182a48b73834597af402fcefeebf41612c2b02292fbea7e2fc5472539	scp/SCPDriver.h
571b480f5be2d97ed4e25d81cf4edbbdf53b0ba5649885b8aea6ba4358f1462d	scp/Slot.cpp
fc4e4332b1d1478d170c4e564eb2e73491a760aaf19aa96100810d84360e39a0	scp/Slot.h
562b9c14685781a06e61ff66d6ab319f1ce97a3c5e563194a6abde5fed395dcd	scp/readme.md
985c26bb250f7d775c31d9a91c7502889d6a233dd3a47139f4eff593a6787b86	simulation/LoadGenerator.cpp
1da9476cd6db05cdb85e9ea760bc98c4eeabc1436c749f8aa59d90a61092d648	simulation/LoadGenerator.h
d61a1df61c30e6642000fbf62264036b6f5eb19f9a76c44e0565a635e9368454	simulation/Simulation.cpp
cf88404b7aa7076c9e13e6c7ccd519cf2c2753820e598d06b46f4750bc9e38e6	simulation/Simulation.h
f8a33fc03bfa8f21c45eceaad1444f9f20e3fc12c0e7828031a4928a2fe8e84e	simulation/Topologies.cpp
0c196fbcd44cf921c310a257f670e4e95bd8c21a5a1fc1600c07719d1bae2be5	simulation/Topologies.h
d66f14a32fe768d08dd87d96d49c50c0183fcd34e2796e0daf9b77db610cf9a8	transactions/AllowTrustOpFrame.cpp
e7d9a363ccc3ffb8d96326acefbbbd3720c392f75c8108df886828e1e88cb553	transactions/AllowTrustOpFrame.h
9def69793a55f940bd70093f659bff6a309a89d884102e3d2a8281258962d683	transactions/BeginSponsoringFutureReservesOpFrame.cpp
01c04a27b9fc7f5153a77fffe47ffa95e3cd5fed9cd4e9c1c71315120b1de74b	transactions/BeginSponsoringFutureReservesOpFrame.h
5314cb951b59096683d07e6a81169eac4df871281b070b8c31c952ad179eaa80	transactions/BumpFootprintExpirationOpFrame.cpp
f3f9f4c33d39b3f5b874214ddbb593a4f89f59462865e67897f7e5d584cd1369	transactions/BumpFootprintExpirationOpFrame.h
db931d8d79040ca1ba1cb65aef7e633bc56eb5827c850abd0dd8ed8aa168e77c	transactions/BumpSequenceOpFrame.cpp
2557550f23a237a8fc8f9ae66945f72dd9feb8231c585c8768f0a8a050191069	transactions/BumpSequenceOpFrame.h
a205768bce506ca7005c946ff8350469dd2c7f6e7425d859fd5ccb4738e37f47	transactions/ChangeTrustOpFrame.cpp
466c6b5d5bdc20fe7e0197c07c476daa75d2dfd41a2007d5fca58570d2bbcbec	transactions/ChangeTrustOpFrame.h
f1363e4b7e08c7d6abecf004d6f55fb223fc21c5138f66909b4b6d41f7ff501b	transactions/ClaimClaimableBalanceOpFrame.cpp
923731c7eca086afd53d1c1f062323512c37bc170f66cdd626650c0e88b062c3	transactions/ClaimClaimableBalanceOpFrame.h
208237742dca0763482edc9cf5d55fdbe7c0590a0bc2486728bd3b637de59c6f	transactions/ClawbackClaimableBalanceOpFrame.cpp
496902584d6b89f88ddfac597d168627a799e013dfb24a58c9d8e207b83262b3	transactions/ClawbackClaimableBalanceOpFrame.h
6e869838d6803088ae55011c62789fcaf4d83a1e46305ce69d77ea268778441a	transactions/ClawbackOpFrame.cpp
182cac627ece3f5a32bb843e42a74188605ae34e03308c6e4e475a65a0444fcc	transactions/ClawbackOpFrame.h
60aa1b48bcf445aa0dd6146862b09c18b3902e65e31f013619a6d34732031b6f	transactions/CreateAccountOpFrame.cpp
9c73f6f56c29570deca5b5af95dd0e8504506cb3fd24a1279b8d3d0f52bda217	transactions/CreateAccountOpFrame.h
bda79b6825be15982a0ac98b45e28c847ae8a5d72148a5a274c0768f940d6b46	transactions/CreateClaimableBalanceOpFrame.cpp
d91204de339b4fdde1e792e7d1f2aeee1582e8a3bc5f8aa476a4dea580cfc1be	transactions/CreateClaimableBalanceOpFrame.h
ae329574abddb48bf559e73e91fb60be1c1f4016e9d12c9f3b90bf08248137c1	transactions/CreatePassiveSellOfferOpFrame.cpp
a97fe14fe2d3ea05791ad9581dd202cae8da38ae93ea10c6c469a09828ee4a14	transactions/CreatePassiveSellOfferOpFrame.h
b85481969eedcf88e3ceebad4bd7c1e6982417c2034214cd0915b5b166908f5b	transactions/EndSponsoringFutureReservesOpFrame.cpp
d06dd353627b39fc8212d0f3a4fdbd6f94b3c632b368aedd0104169bfe454918	transactions/EndSponsoringFutureReservesOpFrame.h
59c9a1eebc4b32dd368abf288019be795af8fb40c47a6d7de9e0c361915b4863	transactions/FeeBumpTransactionFrame.cpp
a4ab50e7ef5a9cf958f54b911ff9fe993b422e843d66e37979f46251199d5482	transactions/FeeBumpTransactionFrame.h
2264e816be8c1467a1e78598d1661c910b1c4e44ab766969fa58f3bd36a641b3	transactions/InflationOpFrame.cpp
fc43132bd6f6013f91629f103f0e487f52fd5472247e30b0d81599d6b4a265fb	transactions/InflationOpFrame.h
dc9266a289d17e3bad1803d0e30125e633363e80074fb32dccd18c42c2bd47e4	transactions/InvokeHostFunctionOpFrame.cpp
2ee9a279337da9ad6b16199aa655f81abbc21c840f1294fe7041f1357c11ee6b	transactions/InvokeHostFunctionOpFrame.h
541f7a1affd315f90801b00a56d1996ed424071f09169f90c0341aa71a66b897	transactions/LiquidityPoolDepositOpFrame.cpp
41ba40f81281db1ab13b9b8d47c9a801c9f29e24a88bfa4c21cab7c936da26f8	transactions/LiquidityPoolDepositOpFrame.h

8690a1de7e46fff07a6624d4259d51f6dbaa8f4e8399014585a8a84a4e002d6b	transactions/LiquidityPoolWithdrawOpFrame.cpp
f873c39a810231aac0fa63f6d6eb76a38eb19e8091cb212044119f7dae71be73	transactions/LiquidityPoolWithdrawOpFrame.h
641e4cbd4a858ec25e514aaf787d9d037dcf3135c0d2bfa41e0f653bd8b75fe4	transactions/ManageBuyOfferOpFrame.cpp
65cdafa9e90675391bde3cd448a4b223d66ffab88f6a2e781546ab9315bd97bc	transactions/ManageBuyOfferOpFrame.h
02d30d540f09ec87f4006cc645e07f60e8f85ae545ee52d5dfae4f30a3b55049	transactions/ManageDataOpFrame.cpp
efe945a94a39fe16c858a1abe819e6b14e6e223bff3a203746b04e7f08a76089	transactions/ManageDataOpFrame.h
dab2ce321951716e7e231ee32ee00c1b1038fe339bb9a4811f786d56ccdba09b	transactions/ManageOfferOpFrameBase.cpp
77caec25f40c168a1be034ed22c66034bd0208996ec696516e23178efaea2098	transactions/ManageOfferOpFrameBase.h
67a108a05784e23907bfda0ee180c2f979d963ff28460f014a5aa7fe8bd0d3f1	transactions/ManageSellOfferOpFrame.cpp
6bef99bb7a5cb465400d38f4eb9a7b78b4aec443d30e18fea04b5e4868ae514a	transactions/ManageSellOfferOpFrame.h
7f6ce6efe16994e491b87dcc862b136ed85c9cefe37af8628a75e340d03119ba	transactions/MergeOpFrame.cpp
7e21ae8dd84e1d5368aa53fc3cb19213e25d3e5224b70f6b54b730b066b62a6e	transactions/MergeOpFrame.h
942023abcbb9d2ec70824993c61dadfb04011bc63ea139b8386ed68e941836d1	transactions/OfferExchange.cpp
06961b6a688750688e11758e1bb0a7bf3eef7fc9e163ee86db789902dd2931d1	transactions/OfferExchange.h
a2834b5b0e87837689fb9922034effc35b9a0f53caff279ae56fba30fe517721	transactions/OperationFrame.cpp
5d6a5115daa7fddf00d3ebfeba617e2b5db7bb787fd4e37edc79f5b8c7fd0bee	transactions/OperationFrame.h
e0576591df30845053bcebaeccc845a296500d9965dad2da99c18a2035ddada5	transactions/PathPaymentOpFrameBase.cpp
449f76364b8cd8f1b2d3cef2dfda3b1c44164403cb4f1ee477179b87e2719ddb	transactions/PathPaymentOpFrameBase.h
10147d2dc067e4456ee36c651f663d6361d89709f271672c2455f3db638f5644	transactions/PathPaymentStrictReceiveOpFrame.cpp
a989a96f1c6993b9cfd579f8d3da4b273aed7862066b681de964fbf8691a9a25	transactions/PathPaymentStrictReceiveOpFrame.h
c0d01fa5f21bc95b09497e3e5f56272e6c86974f0423a7add45d51312b51893a	transactions/PathPaymentStrictSendOpFrame.cpp
9cb6290b27d7f4040ac0c609f8f6d8ae957027f1dbe7191acd953bc4cc0f59d1	transactions/PathPaymentStrictSendOpFrame.h
376ce56e5c25b3a6ef2b2e7df44f6b53a4b260971f3c0090c31cb61b790d5e3c	transactions/PaymentOpFrame.cpp
5b9394366323440ed5d3fc52b999e86fb03c9bc8beb6f8c290eb6813bd1a8601	transactions/PaymentOpFrame.h
c6f3f2c10ad206efd39734a926464a3615a005da8cb9d9d4a07e2fdcd5963317	transactions/RestoreFootprintOpFrame.cpp
8e68ce867b4d6234d6674d5b1d66418075d32d6272e04ab771f9f88e99cee5b0	transactions/RestoreFootprintOpFrame.h
9131b2775372aa82fac68a5cbde5951f012e98464461733d994d73bf34449d46	transactions/RevokeSponsorshipOpFrame.cpp
780c2fbc5407ec90b40d42115a5765d6a0f9088c24d364b99dfe72ca82f03c5c	transactions/RevokeSponsorshipOpFrame.h
1d32dbf56c787e1f8751df9e8be423a2d5bdf7c72458a1fecf19cd979b1e745a	transactions/SetOptionsOpFrame.cpp
0ac21ee7675f1c99803a1912efd81ea83b1588e62a6b9658c57dfe3224fcf0aa	transactions/SetOptionsOpFrame.h
8e01dba7b39e02b2a6b05557c4f2dd34f3144a634fb878af1b8443f38cf8b075	transactions/SetTrustLineFlagsOpFrame.cpp
5ea8b01a603fb5d55c9aeb2cb091e61a925b4406e1eed3fa0fcb9646be45f2f9	transactions/SetTrustLineFlagsOpFrame.h
fe2323a9a4348bf79a2dfb83ac6abc80416b7072e9fd8c63d17ca4ce76821cd1	transactions/SignatureChecker.cpp
a5e6b7c8ca30d5d7d993fda1e3b6b258b942178a08be0eb1b0d03eaa24643323	transactions/SignatureChecker.h
0517985ec0876160e92f2d95a32b9bf7ff160106b5ad9d1a4fb624ee14a86467	transactions/SignatureUtils.cpp
1986a6d29ecae59c4df4f5d30399ce4c0ddfdd6e4f95a58e45d5cdb4ecbd5f1a	transactions/SignatureUtils.h
85e24c4663975dc1f61ae5c55c04590f6614ac36655cb7be0ba2376fff1f7beb	transactions/SponsorshipUtils.cpp
444eeab40cb4f755464df1c897a7517a3d21734ef5c28d7adb8eb0100ba60000	transactions/SponsorshipUtils.h
6af757f266a164ae9093ce264c97e078921b7f76e1241185bdad0606eb5d4f00	transactions/TransactionBridge.cpp
2b4a95a79f61fbec6ecd2e5c5a4537c1c4973b1dcbdb141f251693336d77e63b	transactions/TransactionBridge.h
c240e28118adf68eb13cf0fae4c158fa99872c67f670e2b64b7b69ef5c2ce112	transactions/TransactionFrame.cpp
62252cdd8f57239d9f7ecf72e02e22b579acd8f3713c6a1532ded5c914846213	transactions/TransactionFrame.h
c7895cf0043f8333d50782d7a59bef89c4fefa09aaa2553bbca889ecf7ca91c9	transactions/TransactionFrameBase.cpp
741b3c1aa511b74cf69f72114f69ae6fba08733cf40d98273065402bcbc1a9b6	transactions/TransactionFrameBase.h

fa0db2863671ee4a83b480c4f4521e3d61041b102cee6ab946c903de01fb05c5	transactions/TransactionMetaFrame.cpp
fd3469e1da0f205dac831ba2eeb7f868fc837b961fac358efd62fd597e107b5d	transactions/TransactionMetaFrame.h
46411b4234c3c8f2f61dcb37a01a1c2e4dc5358b0f2b42e2118d2c1eca4d80d6	transactions/TransactionSQL.cpp
a7ab91e017842c67d181f866f510464efe7c35106aef2ab454ec54bd45004873	transactions/TransactionSQL.h
5d860bc38b99bcd7e40530798346fa735920c0163ad355ef8bf22ef64d4ffe5c	transactions/TransactionUtils.cpp
9f3b94ff92069de86bd2485b2a4a561665e833de6efc440c7fe2eb52965318cc	transactions/TransactionUtils.h
feffcf03d0befb5ac02aac80541f6fec2bf1fc5fde8ca51e7e3ed027d8931f9f	transactions/TrustFlagsOpFrameBase.cpp
f28c777aba27622ea623bb37c79cf90f54a53e287cb97d3cf1832230ca60dc99	transactions/TrustFlagsOpFrameBase.h
db0d9384394ebacff139d72cdd346e64ac12042067f3f6cca4de7908670e61ca	transactions/readme.md
f1c05756a5898d01b25fb7af5f5e01806a95ea8057f829a0cfa51927b7f37abd	transactions/simulation/TxSimCreateClaimableBalanceOpFrame.cpp
b865b17db96c24b1ec5b235587e370959ffd39718792ba38fc4b2c8be027e861	transactions/simulation/TxSimCreateClaimableBalanceOpFrame.h
9fee7a950a755a3baa544907645360a3923d6c9a1e392b83a9d3497a703a024a	transactions/simulation/TxSimCreatePassiveSellOfferOpFrame.cpp
193066ce08dcffa8ce07aa7d38bae07017f25395369511c956937324cf8fa1d4	transactions/simulation/TxSimCreatePassiveSellOfferOpFrame.h
785f76634e2b8986785a1a10bdad76c7de382a8d98bfe7fd7a3ae619138db6f1	transactions/simulation/TxSimFeeBumpTransactionFrame.cpp
bb1ef1d382e100e4be6425ecb583cd1cba658406fde31d66c2f85ff24eb9ed2e	transactions/simulation/TxSimFeeBumpTransactionFrame.h
001f3b622accd4c37ac4eb7a3bf56b049d2a24ee2df8b7366efd020ce4cb46fc	transactions/simulation/TxSimGenerateBucketsWork.cpp
9facdec8cab646b1ef2eec6854df45c6b3e4e860f2e4a23a19c26a20090e404d	transactions/simulation/TxSimGenerateBucketsWork.h
4f764b1a696496b711cefa7efa6ceaa182ad0ddbc483bc2e44feba099983cd72	transactions/simulation/TxSimManageBuyOfferOpFrame.cpp
a13df78d1f371b0c5f8d800c33cec6a70acf6194b50fa526dba476395cb8daa6	transactions/simulation/TxSimManageBuyOfferOpFrame.h
d4504b0242c00cbebfc3b2fd60c43459e5b98968b2ffd36cfdf8d32a14d33c0b	transactions/simulation/TxSimManageSellOfferOpFrame.cpp
bde6217ae7f4c19d4c04d1034e53651835b2c0134a171c5ab99e79d06643d1a4	transactions/simulation/TxSimManageSellOfferOpFrame.h
eb794dc814cc5b203d3cdb3c384456cc5e1b4879d4b67a5e320809a7efc628ae	transactions/simulation/TxSimMergeOpFrame.cpp
d9df9819576781b9df843da97cf1729e87e1c15dc81416b5c1b9aac6a58a2b40	transactions/simulation/TxSimMergeOpFrame.h
9bd63019e82e193c72c9a1ad496fada2074573b8fe928bb6775db09da6b97606	transactions/simulation/TxSimScaleBucketlistWork.cpp
9257cd37eab6e0c3044cd49cd30245d40db2ebf72cf24ea7c76dd0031e209f39	transactions/simulation/TxSimScaleBucketlistWork.h
dea760667ff25cf8d4569b88b111ab9c3d1d725c2c1054bbdaeb51ada0ebb7dd	transactions/simulation/TxSimTransactionFrame.cpp
8e800fa4e056784027ea29401484ce24441e9b49eaf44ee0dda6cb6d9cf0459c	transactions/simulation/TxSimTransactionFrame.h
9c2df976bef48f529e042b00b738e4da1ed805f14eb4416083ac930b51d54853	transactions/simulation/TxSimUtils.cpp
08a53276e664a378309f149ece96fb62e9e41b42fe10f8b2876cf8e5614d94f8	transactions/simulation/TxSimUtils.h
ba238dcc837447064426c0bd74654517248fdb49cfcd984131fa83aa33e8ac58	util/Algorithm.h
9388a2aeabc5bc0597cae2ee92ea6abee5fe0da41c4b32f220f72182ace0e004	util/Backtrace.cpp
250de19b323537f8a2726278242c5990eedce73c061ae85a79d513977c3cf677	util/Backtrace.h
8e676c1639cc027873678dc183d09502911b0e8a7e0fe94d8abc5b49a54a0e78	util/BitSet.h
60f4640fceed5aa53454f02c05cdea1be35d3ab5a93d42add7f2264747eb83cc	util/DebugMetaUtils.cpp
fa59d809725522ce627fab67e764a986e9ec1785e56fb182a1080f16db4554e1	util/DebugMetaUtils.h
f4f55590bb2a5e1b2297349b63ef81688b9d99ca5e36a7f8ba02c99be94c979b	util/Decoder.h
40598052424a252803c2e45045018c619f80d45380bbea5b5d5ae7805aa201ac	util/FileSystemException.cpp
5a52e16cff480bb0dc4c5af11d950b64fee718ce51e139984a719920283bb90f	util/FileSystemException.h
a921b878abf75ce02d39d26f5a8451fb5748eaec67086c95f9bc108ce0c30a1e	util/Fs.cpp
8325691680c7425a24e28f71afc58989c1b3c313561502670b1c6cfd83335c41	util/Fs.h

7054d79973248d0592e861b08ec76a58072e69e579140f5c1fa7fcf9a198f00b	util/GlobalChecks.cpp
371519cc6755603ab5db60fe697e130bd16a13213e8ec22216c76f53f2a40a73	util/GlobalChecks.h
aa5729a127d8f25094b35216930e8be41ef8ae68c80030795dc99ae552126f9c	util/HashOfHash.cpp
8d5de5402661411bbf363f3ad9b96ec538e7f4790e44519019169d419cc51657	util/HashOfHash.h
13da2fb1f01550b734a18d645abd5f27ad18b2e846c2a4bde87b45cb90f8202a	util/LogPartitions.def
4056a09b476af11fe7c6f3e021259d9bc31b4bc00c4765d86adc7856a254fc56	util/LogSlowExecution.cpp
bdd4494a4d9b5d7b233f52923e4672309b3083f663739c96ba54661f708f9063	util/LogSlowExecution.h
ce004d6cbc2c791a877dc3e061331181da2c0c1fa337180b0704a23d5afa51e0	util/Logging.cpp
a56ee0be0d880b473cde25ab4b7ef86e464e6b57390e06d3adb1771f61fcf9fe	util/Logging.h
0dfb3cca4c60122c3c390e484bb3a510cd5ddbfc76d140bcd9c53e5a07a929d9	util/Math.cpp
6ddc1c9e466a99097e216d67ff4cf6dca4039a8862fe6760f9645a267ce193ff	util/Math.h
90450888b0a26cf572757b56bb84757fed46cc507bb122543a7390e9ef962f27	util/MetaUtils.cpp
85e9964e9051411a0fcb915d801f00b1aebb5ecf85bc5e4b7c8c9c4a977adc4b	util/MetaUtils.h
c08cd9a9b874c52d69e7f5d20a82cc42ef9cad7bac587448ef8a33e18eb576bf	util/MetricResetter.cpp
88a31d221c5d1a846e23c32a60840d6a8164ff74b41791e202ed2e288af7688d	util/MetricResetter.h
a030b3f7a9cd1d1d63e335a356cc52942da77a954bebc952300f34202d32d11f	util/NonCopyable.h
0d2c7d5f8e3df3dace91a91e0c9fc959c96d2b175428c267d1bfb27d25973cad	util/ProtocolVersion.cpp
347b472182216fe01fa73940604f62681086d6b8df5fa129042f976b373ea382	util/ProtocolVersion.h
24352bfb2d7922ff6a1b39ed49eabf654a2f8fe5fa908000d29485600332987d	util/RandHasher.cpp
ad270125383d9f09c23448ca8a2e746e26ae9ceeab1fa19326a9440589e8a4d2	util/RandHasher.h
6f9464c46acd3a4fb1f222521ae4ab7938afa0fa33ab736dfaf4c567e2659b81	util/RandomEvictionCache.h
590682fc34df3b45d296a401f05e28d3e488a86b6e2f1909f9f0525dce351374	util/Scheduler.cpp
13cfa7c207499b456f2df5ac93f68ec4ffe83521cb322a6b7bedb2548910af3d	util/Scheduler.h
2cce32ec2bfedd0d5b30a514fb593ad1e4947f75f5baef61227c82674264833a	util/SecretValue.cpp
dc5482679000b5c709b600b3201320b477c7d5894bbaaea8a8a49c3e21cee1fd	util/SecretValue.h
927cc0348192537ac46d3325e38de0002c195fe92eee05acb01338003a7506f5	util/SpdlogTweaks.h
d1451fd97d95c3f9d692369cbd1a9f14cab7787c130c90de771e246ab9090382	util/StatusManager.cpp
2efb14a1a204f7e7b2f7d9fa2701990b890197ed7fb8481b4948c637ef5082dd	util/StatusManager.h
6c4c1b7a0734744b27a6dc79f50abaa6e01e4519b1c4cb808c934f41a3bcb039	util/TarjanSCCCalculator.cpp
a7be3b3896e2953cac6e29400f0612923d9fa38122249f09c6f93eb1a9595faa	util/TarjanSCCCalculator.h
163911b71ca69cfe2d11fd6b9b37a85c056ff7fd4d06617ff2c5541bfd60dc31	util/Thread.cpp
79674ec14e1404992771f9b76f20cfe949c224e4fa9c68483d2ffd802018af15	util/Thread.h
c4bff7fa02efd1ff857dc52245a47d73ea82d8afca626480ec533d832bb80e3f	util/Timer.cpp
b15c93907271357 53d64b88f42bb32c6e0bea17d1454c9d74ad9d641a050fc3a	util/Timer.h
a4d2f9c9d9ddef0ac4f391f9fee733cd7a3552bb32dae6927273f1a6f4d193b5	util/TmpDir.cpp
3da3b472f2b616b774c70d2e56ccede149e5e8b9e5d9b050a3d959e9b2acc298	util/TmpDir.h
9a2c5487019585a7221fec88978226f87ffa6fba8159da6acb5a5001379b8afd	util/TxResource.cpp
9f14ed01dab28109d415c1997748ef1f521fcd33ed9a71a452888bb4ef90a7b5	util/TxResource.h
505bf315461a5bf259e0f0b10edc05f2786ab7178bf21edb272c099a20abf6ad	util/UnorderedMap.h
c787ad546293e75c717ccf9da59b20e2fb4634e32eaac5d8d0c546c577a5ebc4	util/UnorderedSet.h
d03e40ae6a1c57a6940ea4ccb4d06ca3249f089de58615b7f6f62cdde623af54	util/XDRCereal.cpp
ff086751df806b72ecbf171cadb42945183e54cfaf29c0ad763fb78ef7d1add0	util/XDRCereal.h
b6ee50b686b8e4423a3702e934521d945b4c9bbdb029aeb8fd2eb5096f97413a	util/XDROperators.h
5a4b6ab35756364052edd44e11faa9d616a309c3336a1cdea1523088d14b3e6d	util/XDRStream.h

d2e8e30df72511af8abb19894a26b4ee297bee0dc510bcbde5866cfb9d24dc15	util/asio.h
5c6c477d442974ecb9b53b1e016eb5d9b3c254db314e79a109c18a04b5a154d5	util/must_use.h
2ad3e55c47c33312f8973aef029e6f228bb6b433122f7f6c4f3285ba0882077f	util/numeric.cpp
cbe56159abb192a82da95ace980624cee4f42163ca83e921ff1b036b26058bf8	util/numeric.h
f50ec7737e11a2a0aaf6d2ec08a2f57007d8e2e47807e22de6925ab97c75c82c	util/numeric128.h
7a2d5b4a82659e45466839975872ef643643254b028579238c4282ee8a8531bf	util/types.cpp
78652aa3e62aa1719b3f2ac992d57317df10cb8294ce896bf6e88a737bb51a4d	util/types.h
94da1ff4cf100c1919d528edf69f97958cc05371c782382c1b1f12132b9d7cb4	util/xdrquery/XDRFieldResolver.h
8a0c7d323cce7bb0c6f54cef9c4bc042fd7ff840291daf9ff6ae888e4bf98616	util/xdrquery/XDRQuery.cpp
020d1cb08f40447931c6b14d1df6fa7b4df14da3371d40039e6ecd8c8ab6f557	util/xdrquery/XDRQuery.h
f5348464b85cd30bbc874aa3319118ac4f6f6ae6aa6c9a9de3f2996b18b52ac8	util/xdrquery/XDRQueryError.h
9134a87c529e2c2d87e7e1ef03c733ce2e996c98e49690de7455851acadad4ac	util/xdrquery/XDRQueryEval.cpp
b18e958d2b97a9b6c1246cd19436496acf80f869823589eacedf4f3dde6a539d	util/xdrquery/XDRQueryEval.h
62b643752b6d8a2f699d94f11bebbe665a6cbbb8fd68d10397a60c3e1ae00ebb	util/xdrquery/XDRQueryParser.yy
b0d7ac770adcc4e2d637b36b2fb8a68f509ae9b9d89b5e3d7ed618b4ab9ba260	util/xdrquery/XDRQueryScanner.ll
525820f519d8839c074cfc717117ed3109d525f62d3f8455a7ec2cd9d096bbfe	work/BasicWork.cpp
7ff4e708e36185e0dc9384af5a9defc10834c5536fcc6e9fb2a0c58d9d61f0e7	work/BasicWork.h
a18a8ba07bbf1073499761367698bbaff2c63be0d2bacd7b0dd6edbd6bbd762d	work/BatchWork.cpp
a6652bb423223adf917fc094177eb7df31851d7feaf9401ff5e534d4da552da8	work/BatchWork.h
c37683eaf6154477445d8845e1ac33ecef66aaea9283ca718d55e8e7e32ec0d7	work/ConditionalWork.cpp
21187b2e50a8bf29be4e072d50438bc3a545bee4198ce8708536fbdcdafe0871	work/ConditionalWork.h
b83ea990b317d14aa055a857682dbf337c127d618b2f5dd84e4c103ef4cc0785	work/Work.cpp
7a1e2589bcdd2adb2cf4342149e1eaeeef374355879e19e0884a2d094d0036ed	work/Work.h
db4b68ec8286af16adce8f3234603a93398d017940da3ffdb3269850570e41ba	work/WorkScheduler.cpp
ee1eb97e0d80744cb416329f754a2af24319d23e34867a4e7b395f73bfd64745	work/WorkScheduler.h
fe6c0463101f1f7b85ee5dfdd47841dee3f57a3c18644352c8bad2ff6f71393c	work/WorkSequence.cpp
c613f94426dc8c433f1f6f2992881a37c1bd43210d4d7d1ebaf3fe7f21e26a03	work/WorkSequence.h
a2648a083d6d50a9e94e7ca966c3c1161811868ae4607b0b275e465633df1ac0	work/WorkWithCallback.cpp
6f7f66a8770167ad16864fd8678a0782c461f161f4e230e574b95e3e8af22982	work/WorkWithCallback.h

Summary

Overall, Elar's codebase exhibits a well-designed and organized architecture that prioritizes efficiency, security, and maintainability. Its adherence to the Stellar Consensus Protocol, modular design, and efficient ledger management contribute to its robustness and reliability in maintaining a distributed cryptographic ledger. This audit provides confidence in the system's ability to fulfill its role in the Stellar ecosystem. The code review and auditing process have uncovered several key areas for improvement within the C++ codebase.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io