3.5B
TOTAL SUPPLY

24
SMART CONTRACTS

883
TEST CASES

SBET

WHITEPAPER V2.0

Decentralized Sports Betting Protocol

Peer-to-Peer Trading, Pool Betting, NFT Staking & Prediction Markets on Ethereum

March 2026

Token: SBET (ERC-20)	**Chain:** Ethereum Mainnet
Audit: Versus Security	**License:** BUSL-1.1

Contract: 0x2ed2cc2c858a8a8219fd2f2d9e170285dbd02756

Website: sbettoken.org, sbetchain.ai

sbettoken.org
Public Document

Table of Contents

CHAPTER 1

1. Executive Summary

SBET Protocol is a fully decentralized sports betting platform built on Ethereum, offering four distinct betting primitives — peer-to-peer trading, pool betting, NFT staking, and prediction markets — all governed by immutable smart contracts with no centralized counterparty risk.

Unlike traditional sportsbooks that act as the house, SBET removes the intermediary entirely. Users bet directly against each other through EIP-712 signed orders matched on-chain, or participate in pari-mutuel pools where odds are determined purely by market forces. Every operation — from order execution to fee distribution — is enforced by audited Solidity code running on Ethereum.

3.5B	24	25K+	883
FIXED TOKEN SUPPLY	SMART CONTRACTS	LINES OF SOLIDITY	TEST CASES

THE PROBLEM

The global sports betting market exceeds $500 billion annually, yet it remains dominated by centralized platforms that suffer from fundamental issues:

- **Counterparty risk:** Users must trust operators to honor payouts and not manipulate odds
- **Opaque fee structures:** Traditional sportsbooks embed 5-15% margins (vig) with no transparency
- **Geographic restrictions:** Regulatory fragmentation locks out billions of potential users
- **Settlement delays:** Withdrawals often take days and require KYC friction
- **No composability:** Bets are siloed — positions cannot be traded, hedged, or used as collateral

THE SBET SOLUTION

SBET addresses every dimension of this problem through a comprehensive protocol design:

TRUSTLESS EXECUTION

Smart contracts enforce all rules — no operator can withhold funds, change odds after placement, or deny payouts. Code is law.

TRANSPARENT FEES

Protocol fees are on-chain and immutable. Users always know the exact cost structure before placing a bet.

GLOBAL ACCESS

Anyone with an Ethereum wallet can participate. No KYC gates, no geo-blocking, no withdrawal queues.

COMPOSABLE POSITIONS

P2P orders are tradeable before settlement. NFT bets represent positions as tokens. DeFi-native from day one.

KEY DIFFERENTIATORS

FEATURE	TRADITIONAL SPORTSBOOKS	EXISTING CRYPTO BETTING	SBET PROTOCOL
Counterparty	House (operator)	House or AMM	PEER-TO-PEER

Transparency	Opaque	Partial	FULLY ON-CHAIN
Betting Modes	Fixed odds	1-2 modes	4 PRIMITIVES
Fee Transparency	Hidden vig	Variable	ON-CHAIN, FIXED
Settlement	Days	Hours	INSTANT (ON-CHAIN)
Position Trading	Not possible	Rare	EIP-712 NATIVE
Security Audit	N/A	Varies	146-FINDING AUDIT

> **Investment Thesis:** SBET is building the decentralized infrastructure layer for sports betting — the "Uniswap of sports markets." With 24 production-grade smart contracts, a completed security audit, and exchange listings on Coinbase, Gate.io, and LBank, the protocol is positioned to capture share from a $500B+ market transitioning on-chain.

PROTOCOL MILESTONES ACHIEVED

MILESTONE	DATE	SIGNIFICANCE
Token Launch	2022	SBET ERC-20 deployed on Ethereum mainnet with 3.5B fixed supply
Exchange Listings	2023-2024	Listed on Coinbase, Gate.io, and LBank — establishing Tier 1 liquidity
Protocol v2.0	2025	Complete rewrite: 24 contracts, 25K+ LoC, 4 betting primitives, treasury system
Security Audit	Feb 2026	Versus Security audit: 146 findings identified and resolved, 883 test cases
Whitepaper v2.0	Mar 2026	Comprehensive protocol documentation reflecting current architecture

WHY SBET MATTERS

The convergence of three macro trends creates an unprecedented opportunity for SBET:

1. **Regulatory tailwinds:** Sports betting is being legalized globally at an accelerating pace, creating a massive legal market that was previously underground.
2. **DeFi maturity:** The DeFi ecosystem has proven that complex financial operations can run trustlessly on-chain. Sports betting is the next major vertical to be decentralized.
3. **User demand:** Millions of crypto-native users are actively seeking permissionless, transparent alternatives to centralized platforms that restrict access and manipulate odds.

SBET sits at the intersection of these trends with a fully built, audited, and listed protocol ready to capture this market opportunity.

CHAPTER 2

2. Market Opportunity

GLOBAL SPORTS BETTING MARKET

The global sports betting market is one of the largest entertainment verticals in the world, and it is accelerating due to regulatory liberalization, mobile adoption, and the convergence of sports and technology.

$500B+	10.3%	$150B+	38
GLOBAL MARKET (2025)	ANNUAL GROWTH (CAGR)	ONLINE SEGMENT	US STATES LEGAL

MARKET DRIVERS

REGULATORY TAILWINDS

Post-PASPA repeal (2018), US states are legalizing at record pace. Europe and Asia-Pacific continue expanding regulated frameworks.

MOBILE-FIRST ADOPTION

Over 70% of bets are now placed via mobile devices. Crypto wallets provide a natural extension of this mobile-native behavior.

DEFI CONVERGENCE

DeFi protocols have proven that financial primitives can run trustlessly on-chain. Sports betting is the next major vertical to be decentralized.

CRYPTO NATIVE USERS

Hundreds of millions of crypto holders represent a massive addressable market that prefers permissionless, transparent platforms.

COMPETITIVE LANDSCAPE

PLATFORM	MODEL	CHAIN	BETTING MODES	AUDIT	TOKEN
CENTRALIZED (CEFI)					
DraftKings / FanDuel	House book	N/A	Fixed odds	N/A	Equity
Stake.com	House book (crypto)	N/A	Fixed odds	N/A	None
DECENTRALIZED (DEFI)					
Polymarket	CLOB	Polygon	Binary markets	Yes	None
Azuro	Liquidity pool	Polygon, Gnosis	Fixed odds	Yes	AZUR
Overtime Markets	AMM	Optimism, Base	AMM odds	Yes	THALES
SBET Protocol	**P2P + Pools + NFT + LMSR**	**Ethereum**	**4 primitives**	**146 findings**	**SBET**

> **SBET's Edge:** No competing protocol offers all four betting primitives (P2P trading, pool betting, NFT staking, prediction markets) within a unified, audited smart contract system. SBET's multi-modal approach means it can serve

casual bettors (pools), sophisticated traders (P2P), collectors (NFT bets), and speculators (prediction markets) — all within a single token economy.

THE DECENTRALIZED BETTING THESIS

Traditional sportsbooks face structural constraints that decentralized protocols can exploit:

LICENSING COSTS

Traditional operators spend $10-50M per jurisdiction for licensing. SBET smart contracts deploy once and serve globally — no per-market licensing overhead.

CUSTOMER ACQUISITION

CeFi sportsbooks spend $500-1000 per acquired customer via advertising. SBET's integrator model distributes acquisition costs across an ecosystem of partners.

SETTLEMENT INFRASTRUCTURE

Traditional platforms maintain complex payment processing with banks, card networks, and KYC providers. SBET settles instantly via Ethereum with zero payment infrastructure cost.

TRUST & TRANSPARENCY

Centralized platforms periodically face accusations of odds manipulation and slow payouts. SBET's on-chain operations are fully verifiable and trustless.

SPORTS COVERAGE STRATEGY

SBET's sports taxonomy system supports a comprehensive range of sports and leagues across multiple categories:

CATEGORY	SPORTS	MARKET TYPES
Football / Soccer	Premier League, La Liga, Bundesliga, Serie A, Ligue 1, Champions League, World Cup	H2H, Spread, Total, BTTS, Double Chance, Over/Under 2.5
American Sports	NFL, NBA, MLB, NHL, MLS, NCAA	H2H, Spread, Total
Combat Sports	UFC, Boxing	H2H, Method of Victory
Racket Sports	Tennis (ATP, WTA), Table Tennis	H2H, Set Spread
Motorsport	F1, NASCAR, MotoGP	Winner, Podium
Esports	CS2, Dota 2, League of Legends, Valorant	H2H, Map Winner
Other	Cricket, Rugby, Golf, Cycling	H2H, various

TOTAL ADDRESSABLE MARKET

SBET targets three converging markets:

SEGMENT	SIZE (2025)	GROWTH	SBET CAPTURE STRATEGY
Online Sports Betting	$150B+	12% CAGR	Trustless P2P alternative to house-book model
Prediction Markets	$50B+	25% CAGR	LMSR-backed markets with on-chain settlement

DeFi Derivatives	$80B+ TVL	20% CAGR	Composable bet positions, tradeable on secondary markets

CHAPTER 3

3. Protocol Overview

SBET Protocol is a modular, upgradeable smart contract system that provides the complete infrastructure for decentralized sports betting. The protocol is organized around six core primitives, each implemented as isolated, audited contract modules.

SIX BETTING PRIMITIVES

P2P TRADING

EIP-712 signed orders matched on-chain. Makers set prices, takers fill. No AMM needed.

POOL BETTING

Pari-mutuel pools where odds adjust dynamically based on total stakes per outcome.

NFT BETTING

Stake ERC-721/1155 NFTs on match outcomes. NFTs escrowed in vault until settlement.

PREDICTION MARKETS

LMSR-backed markets for multi-outcome events with automated market making.

TREASURY

Hub-and-spoke system with 7 specialized modules for protocol fund management.

INTEGRATOR HUB

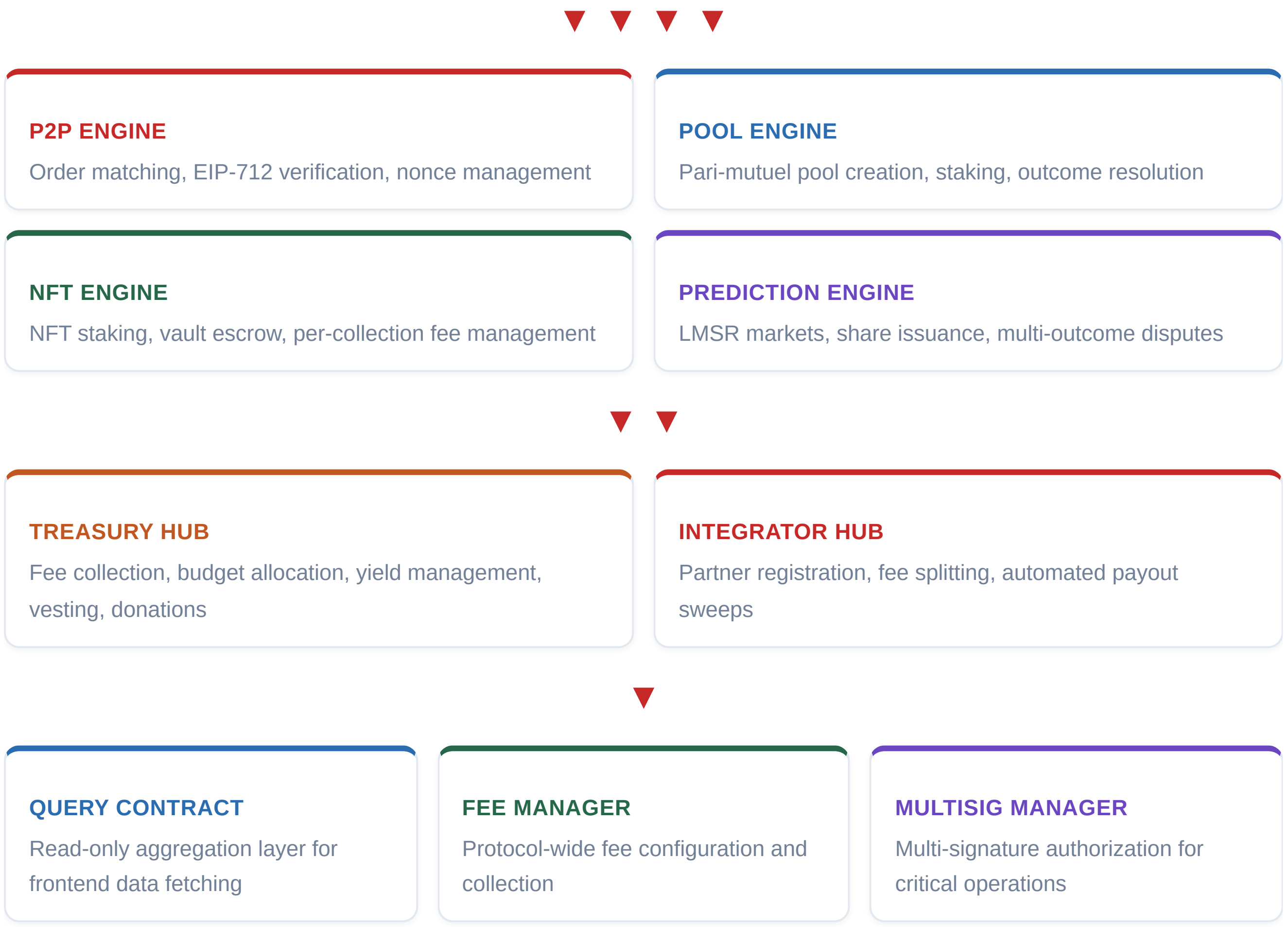
Self-service partner registration with automated fee sharing and payout sweeps.

DESIGN PHILOSOPHY

PRINCIPLE	IMPLEMENTATION
Security First	Every contract audited by Versus Security. 146 findings resolved. 883 test cases. Slither + Echidna analysis.
Modularity	Each betting primitive is a standalone contract. Treasury uses hub-and-spoke. No god-contracts.
Gas Efficiency	Batch operations for pool betting, nonce ranges for order cancellation, minimal storage writes.
No Admin Keys	Emergency pause with time-bounded recovery. No unilateral fund access. Multisig for critical ops.
Composability	EIP-712 orders are portable. NFT positions are standard tokens. Prediction shares are tradeable.
Fail-Closed	Unknown states revert. Invalid inputs rejected. Emergency mode locks all mutations.

PROTOCOL ARCHITECTURE

SBET Protocol — Contract Architecture



P2P ENGINE
Order matching, EIP-712 verification, nonce management

POOL ENGINE
Pari-mutuel pool creation, staking, outcome resolution

NFT ENGINE
NFT staking, vault escrow, per-collection fee management

PREDICTION ENGINE
LMSR markets, share issuance, multi-outcome disputes

TREASURY HUB
Fee collection, budget allocation, yield management, vesting, donations

INTEGRATOR HUB
Partner registration, fee splitting, automated payout sweeps

QUERY CONTRACT
Read-only aggregation layer for frontend data fetching

FEE MANAGER
Protocol-wide fee configuration and collection

MULTISIG MANAGER
Multi-signature authorization for critical operations

CONTRACT INTERACTION FLOW

A typical P2P bet follows this lifecycle:

Maker signs EIP-712 order → **Taker submits to SBET Core** → **On-chain signature verification** → **Tokens escrowed** → **Match finalized by oracle** → **Winner claims payout**

SYSTEM STATE MANAGEMENT

The SBET Core contract maintains several system-wide state variables tracked via the `SystemStatus` struct:

```
struct SystemStatus {
    bool isPaused;              // Global pause flag
    bool isEmergency;           // Emergency mode active
    uint256 emergencyEndTime;   // Auto-resume timestamp
    address treasury;           // Treasury contract address
    uint256 poolCount;          // Total pools created
}
```

MATCH MANAGEMENT

Matches are the atomic unit of betting in the SBET Protocol. Each match represents a real-world sporting event and tracks:

FIELD	TYPE	DESCRIPTION
`matchId`	uint256	Unique on-chain identifier for the match
`finalized`	bool	Whether the outcome has been determined by the oracle
`finalPrice`	uint256	Settlement price (determines P2P order outcomes)
`sideA / sideB`	address	Metadata identifiers for the two competing sides
`oracle`	address	Authorized address that can finalize this match
`finalizationTimelock`	uint256	Delay period between oracle submission and payout availability
`nftCount`	uint256	Number of NFTs staked on this match

DATA ARCHITECTURE: FRONTEND-TO-CONTRACT MAPPING

SBET follows a strict one-screen-per-contract pattern for the dApp:

STEP	ARTIFACT	LOCATION
1. ABI	Exported const array	`src/abi/<Contract>.ts`
2. Hooks	Wagmi read/write wrappers	`src/hooks/use<Contract>.ts`
3. Page	React component	`src/pages/<Contract>Page.tsx`
4. Address	Per-chain address	`src/config/contracts.ts`
5. Route	URL path + nav entry	`src/App.tsx` + `Sidebar.tsx`

This pattern ensures every new contract integration follows a predictable, auditable structure that reduces the risk of frontend bugs and makes the codebase navigable for new contributors.

EVENT ARCHITECTURE

SBET emits structured events for every state-changing operation, enabling off-chain indexing, analytics, and audit trails:

BETTING EVENTS

`OrderFilled` , `OrderCancelled` , `PoolCreated` , `PoolStaked` , `PoolFinalized` , `Claimed`

NFT EVENTS

`NFTStaked` , `NFTClaimed` , `CollectionBlacklisted` , `FeeUpdated`

TREASURY EVENTS

`FeeCollected` , `BudgetAllocated` , `VestingCreated` , `EmergencyPaused`

SYSTEM EVENTS

`MatchCreated` , `MatchFinalized` , `OracleUpdated` , `IntegratorRegistered`

CHAPTER 4

4. Technical Architecture

COMPILER & TOOLCHAIN

COMPONENT	VERSION	PURPOSE
Solidity	0.8.34	Smart contract language with overflow protection, custom errors
Foundry (Forge)	Latest	Compilation, testing, deployment, gas profiling
OpenZeppelin	v5.x	Battle-tested base contracts (ERC-20, ReentrancyGuard, Pausable)
Slither	Latest	Static analysis for vulnerability detection
Echidna	Latest	Fuzzing for invariant testing

CONTRACT HIERARCHY

The protocol follows a layered architecture with clear separation of concerns:

LAYER	CONTRACTS	RESPONSIBILITY
ENTRY LAYER		
Core	SBET.sol	Main entry point, EIP-712 domain, order routing, match management
Query	SBETQuery.sol	Read-only aggregator for UI data (no state mutations)
BETTING LAYER		
P2P	SBET.sol (internal)	EIP-712 order matching, signature verification, nonce tracking
Pools	SBET.sol (internal)	Pari-mutuel pool creation, staking, payout calculation
NFT	NFTVault.sol, NFTManager.sol, NFTFeeManager.sol	NFT escrow, staking, per-collection fee management
Prediction	PredictionMarket.sol, PredictionAMM.sol	LMSR markets, automated pricing, dispute resolution
TREASURY LAYER		
Hub	Treasury.sol, TreasuryFacade.sol	Central fund management, access control, external interface
Modules	FeeManager, BudgetManager, VestingManager, YieldManager, MultisigManager, DonationManager	Specialized financial operations with daily limits and approvals

INTEGRATION LAYER		
Partners	IntegratorHub.sol	Self-service registration, fee splitting, payout automation

INHERITANCE TREE

```
OpenZeppelin Contracts v5
├── ERC20                    → SBET Token
├── ReentrancyGuard          → SBET Core, Treasury, NFTVault
├── Pausable                 → SBET Core, Treasury
├── ERC721Holder             → NFTVault
├── ERC1155Holder            → NFTVault
├── EIP712                   → SBET Core (domain separator)
└── AccessControl            → Treasury, IntegratorHub

SBET Protocol
├── SBET.sol                 → Core entry + P2P + Pools
│   ├── EIP712 domain management
│   ├── Order lifecycle (create, fill, cancel)
│   ├── Pool lifecycle (create, stake, finalize)
│   ├── Match management (create, finalize, oracle)
│   └── Nonce management (min, cancel, range)
├── NFTVault.sol             → ERC-721/1155 escrow
├── NFTManager.sol           → NFT bet coordination
├── NFTFeeManager.sol        → Per-collection fees
├── PredictionMarket.sol     → LMSR market lifecycle
├── PredictionAMM.sol        → Automated market maker
├── Treasury.sol             → Hub contract
├── TreasuryFacade.sol       → External interface
├── FeeManager.sol           → Fee configuration
├── BudgetManager.sol        → Budget allocation
├── VestingManager.sol       → Token vesting
├── YieldManager.sol         → Yield strategies
├── MultisigManager.sol      → Multi-sig authorization
├── DonationManager.sol      → Community donations
├── IntegratorHub.sol        → Partner integration
└── SBETQuery.sol            → Read-only queries
```

SECURITY PRIMITIVES

REENTRANCY GUARDS

All external-facing functions use OpenZeppelin's `nonReentrant` modifier. CEI (Checks-Effects-Interactions) pattern enforced throughout.

EIP-712 TYPED SIGNING

Orders are signed off-chain with domain-bound typed data. Domain includes `chainId`, `verifyingContract`, and a keccak256 salt for replay protection.

EMERGENCY PAUSE

Global pause halts all mutations. Emergency mode is time-bounded — the protocol auto-resumes after the emergency window expires.

NONCE MANAGEMENT

Three-tier nonce system: minimum nonce (bulk cancel), per-nonce cancel (selective), and auto-increment for ordering.

EIP-712 DOMAIN CONFIGURATION

```
EIP712_DOMAIN = {
  name: "Sports BET (SBET) Protocol",
  version: "1.0",
  chainId: <dynamic>,
  verifyingContract: <SBET Core address>,
  salt: keccak256("53019554eadfcd6f3ee53ec6d0bcd18db643b20cfcf8655aae83229cdeb079bd")
}

Order Type = {
  maker: address, taker: address, token: address,
  matchId: uint256, amount: uint256, price: uint256,
  direction: uint256, expiry: uint256, timestamp: uint256,
  orderGroup: uint256, nonce: uint256,
  isNFT: bool, nftAddress: address, nftId: uint256
}
```

Security Note: The EIP-712 salt is derived from a hardcoded seed and must match the constructor parameter in `SBET.sol`. Any mismatch will cause all order signatures to be invalid, preventing unauthorized replay across domains.

GAS OPTIMIZATION TECHNIQUES

TECHNIQUE	SAVINGS	IMPLEMENTATION
Off-chain signing	~100% for order creation	EIP-712 signatures are free — no on-chain transaction until fill
Batch operations	~40-60% per item	Multiple pool stakes, fills, or claims in a single transaction
Minimal storage writes	~20K gas per slot saved	Hot storage for frequently accessed data; events for historical data
Nonce ranges	~95% for bulk cancel	`setMinNonce()` invalidates thousands of orders in 1 tx
Custom errors	~200 gas per revert	Solidity 0.8.x custom errors instead of require strings
Tight packing	Storage slot savings	Struct fields ordered to minimize storage slots used

FRONTEND PROVIDER STACK

The SBET dApp uses a modern React/TypeScript stack with a carefully layered provider architecture:

```
WagmiProvider (wallet connection + contract reads/writes)
  └─ QueryClientProvider (TanStack Query, 10s stale time)
       └─ RainbowKitProvider (wallet modal UI)
            └─ App (routing + pages)
```

This stack provides:

- **WagmiProvider:** React hooks for wallet connection, contract reads (`useReadContract`), and writes (`useWriteContract`)
- **TanStack Query:** Automatic caching, background refetching, and optimistic updates with a 10-second stale time

- **RainbowKit:** Pre-built wallet connection modal supporting MetaMask, WalletConnect, Coinbase Wallet, and 50+ wallets

TYPE SYSTEM ARCHITECTURE

SBET maintains a strict type system that mirrors Solidity types in TypeScript:

PROTOCOL TYPES (INDEX.TS)

Direct Solidity mirrors: `BetSide` , `OrderDirection` , `Order` , `PoolInfo` , `MatchDetails` , `NFTStake` , `SystemStatus` , `NonceStatus`

DOMAIN TYPES (SBET.TYPES.TS)

Frontend models: `Match` , `Team` , `BetSlipItem` , `MatchOdds` , `MarketKind` , `AppConfig` — optimized for UI rendering

ZOD SCHEMAS (SBET.SCHEMAS.TS)

Runtime validation: `TaxonomySchema` , `SportsMenuSchema` , `ConfigSchema` — validate all external data at module boundaries

CONTRACT ABIS (SRC/ABI/)

Type-safe ABI constants for each contract, enabling TypeScript-checked contract interactions via Wagmi hooks

SUPPORTED ODDS FORMATS

The frontend supports seven international odds formats to serve a global user base:

FORMAT	REGION	EXAMPLE (60% IMPLIED)
Decimal	Europe, Australia	1.67
American	United States	-150
Fractional	United Kingdom	2/3
Probability	Universal	60%
Hong Kong	Hong Kong, Asia	0.67
Indonesian	Indonesia	-1.50
Malay	Malaysia	0.67

CHAPTER 5

5. P2P Trading Engine

The peer-to-peer trading engine is SBET's flagship betting primitive. It enables users to create and fill orders against each other without any intermediary, automated market maker, or liquidity pool. Prices are set by the market — makers offer odds, and takers accept them.

HOW IT WORKS

1. Maker creates order → 2. Signs with EIP-712 → 3. Order shared off-chain → 4. Taker fills on-chain → 5. Tokens escrowed → 6. Settlement

ORDER STRUCTURE

Every P2P order contains 14 fields, all included in the EIP-712 typed data hash:

FIELD	TYPE	DESCRIPTION
`maker`	address	Order creator's Ethereum address
`taker`	address	Specific counterparty (or `address(0)` for open orders)
`token`	address	ERC-20 token used for the bet (typically SBET)
`matchId`	uint256	On-chain match identifier
`amount`	uint256	Stake amount in token base units
`price`	uint256	Implied probability (0–10000 basis points, i.e., 0–100%)
`direction`	uint256	0 = Long (Side A wins), 1 = Short (Side B wins)
`expiry`	uint256	Unix timestamp after which the order is invalid
`timestamp`	uint256	Order creation time (prevents stale replay)
`orderGroup`	uint256	Groups related orders for batch cancellation
`nonce`	uint256	Unique per-user incrementing nonce
`isNFT`	bool	Whether this order involves an NFT stake
`nftAddress`	address	NFT contract address (if `isNFT` is true)
`nftId`	uint256	NFT token ID (if `isNFT` is true)

PRICE MODEL

Prices in SBET are expressed as basis points (0–10000), representing the implied probability of the outcome:

```
Price = Implied Probability × 10,000

Maker's Stake = amount
Taker's Stake = amount × (10,000 − price) ÷ price

Winner's Payout = Maker's Stake + Taker's Stake − Protocol Fee
```

Example: A maker creates an order at price 6000 (60% implied probability) with 100 SBET. The taker must put up 100 × (10000 − 6000) ÷ 6000 = 66.67 SBET. If the maker's side wins, they receive 166.67 SBET minus the protocol fee.

NONCE MANAGEMENT

SBET implements a three-tier nonce system for flexible order management:

MECHANISM	FUNCTION	USE CASE
Minimum Nonce	`setMinNonce(n)`	Bulk-cancel all orders with nonce < n. One tx cancels thousands of orders.
Per-Nonce Cancel	`cancelNonce(n)`	Cancel a specific order by its nonce without affecting others.
Auto-Increment	Internal	Each filled order consumes its nonce, preventing double-fills.

FEE SEPARATION

P2P trading fees are separated into distinct categories:

- **Protocol fee:** Collected by the SBET treasury on each filled order
- **Integrator fee:** Optional fee split for front-end operators who source the order
- **NFT fee:** Additional fee when NFT collateral is involved (per-collection configurable)

Gas Optimization: Orders are signed off-chain (free for makers) and only consume gas when filled. This means creating and canceling orders costs nothing — only successful fills incur gas costs.

ORDER DIRECTION & SETTLEMENT

SBET orders use a binary direction model mapped to match sides:

DIRECTION	VALUE	MEANING	WINS WHEN
Long	0	Betting on Side A	Side A wins (finalPrice favors A)
Short	1	Betting on Side B	Side B wins (finalPrice favors B)

At settlement, the oracle sets a `finalPrice` for the match. Orders are resolved based on whether the final price exceeds or falls below the order's price point, enabling continuous-price settlement rather than simple binary outcomes.

SECURITY CONSIDERATIONS

REPLAY PROTECTION

Every order includes `chainId` in the EIP-712 domain and a unique `nonce`. Filled nonces are permanently consumed. Cross-chain replay is impossible due to the domain-bound salt.

ORDER EXPIRY

The `expiry` field ensures stale orders cannot be filled. Makers should set short expiry windows (minutes to hours) for volatile markets.

TAKER SPECIFICATION

Setting `taker` to a specific address creates a private order that only the designated counterparty can fill, preventing front-running by MEV bots.

ORDER GROUPS

The `orderGroup` field enables batch operations — cancel all orders in a group with a single transaction, useful for market makers managing many orders.

MARKET MAKER INTEGRATION

The P2P engine is designed to support professional market makers:

- **Free order creation:** Off-chain signing means market makers can maintain thousands of live orders without gas costs
- **Efficient cancellation:** `setMinNonce()` cancels all orders below a threshold in one transaction — ideal for rapid market repositioning
- **Open taker field:** Setting `taker = address(0)` creates open orders visible to all participants
- **Price granularity:** Basis point pricing (0-10,000) provides 0.01% price precision
- **Batch fills:** Multiple orders can be filled in a single transaction for gas efficiency

CHAPTER 6

6. Pool Betting

Pool betting (pari-mutuel) is SBET's most accessible betting primitive. Users stake tokens on outcomes, and payouts are determined by the total pool distribution — there is no counterparty and no fixed odds. The protocol simply collects a small fee and distributes the rest to winners proportionally.

PARI-MUTUEL MODEL

In pari-mutuel betting, the odds are not set by a bookmaker. Instead, they are determined entirely by the ratio of stakes across outcomes:

$$\text{Payout}_{\text{winner}} = (\text{Stake}_{\text{winner}} \div \text{Total Winning Stakes}) \times \text{Total Pool} \times (1 - \text{Fee Rate})$$

This means odds continuously adjust as new stakes come in, naturally reflecting the collective wisdom of all participants.

POOL LIFECYCLE

1. Pool Created → 2. Staking Open → 3. Event Starts (staking closes) → 4. Oracle Finalizes → 5. Winners Claim

POOL STATE

Each pool tracks the following on-chain state:

FIELD	TYPE	DESCRIPTION
`poolId`	uint256	Unique pool identifier
`finalized`	bool	Whether the outcome has been determined
`token`	address	ERC-20 token used for stakes
`totalStaked`	uint256	Sum of all stakes across all outcomes
`outcome`	uint256	Winning outcome index (set at finalization)
`numOutcomes`	uint256	Number of possible outcomes (2+ supported)
`outcomeStakes[]`	uint256[]	Array of total stakes per outcome

WORKED EXAMPLE

Example: Arsenal vs Chelsea — Pool with 2 outcomes

Stakes: Arsenal backers stake 10,000 SBET. Chelsea backers stake 5,000 SBET.
Total Pool: 15,000 SBET. **Fee:** 2% = 300 SBET.
Net Pool: 14,700 SBET.

If Arsenal wins: Each Arsenal backer receives (their stake ÷ 10,000) × 14,700 SBET.
A user who staked 1,000 SBET on Arsenal receives: (1,000 ÷ 10,000) × 14,700 = **1,470 SBET** (47% profit).

If Chelsea wins: Each Chelsea backer receives (their stake ÷ 5,000) × 14,700 SBET.
A user who staked 1,000 SBET on Chelsea receives: (1,000 ÷ 5,000) × 14,700 = **2,940 SBET** (194% profit).

MULTI-OUTCOME POOLS

SBET pools support arbitrary numbers of outcomes, not just binary (win/lose). This enables:

- **Three-way pools:** Home win, Draw, Away win (standard football/soccer)
- **Tournament pools:** Multiple teams or players competing
- **Score range pools:** Betting on score brackets (0-1, 2-3, 4+)

BATCH OPERATIONS

For gas efficiency, SBET supports batch staking — users can stake on multiple outcomes or multiple pools in a single transaction. This is particularly useful for:

- Hedging positions across outcomes
- Placing bets on multiple matches simultaneously
- Integrator platforms that batch user orders

POOL VS P2P: WHEN TO USE EACH

CRITERIA	POOL BETTING	P2P TRADING
Best for	Casual bettors, popular events	Sophisticated traders, niche markets
Odds determination	Market-driven (pool ratio)	Maker-set (offer/accept)
Liquidity needed	Pool participants	Counterparty per order
Complexity	Simple (stake and wait)	Advanced (sign, set price, manage nonces)
Position trading	Not supported	Full secondary market

POOL FINALIZATION & ORACLE INTEGRATION

Pool outcomes are determined by oracle-submitted finalization data. The finalization process includes built-in safety mechanisms:

SAFETY MECHANISM	DESCRIPTION
Finalization Timelock	After oracle submits an outcome, a configurable timelock delay allows community verification before payouts become claimable
Oracle Authorization	Only whitelisted oracle addresses can finalize matches. Oracle changes require multisig approval.
Immutable Outcomes	Once finalized and past the timelock, outcomes cannot be changed. This prevents post-hoc manipulation.
Refund on Cancel	If a match is cancelled (e.g., weather postponement), all stakes are returned in full — no fees charged.

USER POOL DETAILS

Users can query their position in any pool via the `SBETQuery` contract:

```
struct UserPoolDetails {
    uint256 totalStake;         // User's total stake across all outcomes
    uint256[] outcomeStakes;    // Stake per outcome
    uint256 potentialPayout;    // Maximum payout if user's outcome wins
}
```

This data is consumed by the frontend to display real-time potential returns as odds shift with incoming stakes.

CHAPTER 7

7. NFT Betting

SBET's NFT betting primitive enables users to stake ERC-721 and ERC-1155 non-fungible tokens on match outcomes. This opens sports betting to the NFT community, allowing collectors to put their digital assets to work without selling them — if they win, they get their NFT back plus token winnings.

HOW NFT BETTING WORKS

1. User approves NFT → 2. NFT escrowed in Vault → 3. Matched with counterparty → 4. Match finalizes → 5. Winner claims NFT + tokens

ARCHITECTURE

NFT betting involves three specialized contracts working together:

NFTVAULT.SOL

Secure escrow for ERC-721 and ERC-1155 tokens. Implements `ERC721Holder` and `ERC1155Holder` for safe receipt. Only the NFTManager can withdraw.

NFTMANAGER.SOL

Coordinates NFT bet lifecycle: staking, matching, settlement. Tracks `NFTStake` records with owner, token ID, side, and claim status.

NFTFEEMANAGER.SOL

Per-collection fee configuration. Allows different fee rates for high-value vs standard collections. Manages collection blacklists.

NFT STAKE RECORD

```
struct NFTStake {
    address owner;         // Original NFT owner
    address nftAddress;    // ERC-721/1155 contract
    uint256 nftId;         // Token ID
    BetSide side;          // SideA (1) or SideB (2)
    bool claimed;          // Whether payout was collected
}
```

PER-COLLECTION FEES

The NFTFeeManager allows the protocol to set different fee rates for different NFT collections:

COLLECTION TYPE	EXAMPLE	FEE RATE	RATIONALE
Blue-chip	High-value collections	1-2%	Lower fees attract high-value stakes
Standard	Mid-tier collections	2-3%	Default fee tier
Unverified	New/unknown collections	3-5%	Higher fee compensates for valuation risk
Blacklisted	Malicious/spam	Blocked	Cannot be staked at all

COLLECTION BLACKLIST

The protocol maintains a blacklist of NFT collections that cannot participate in betting. This protects users from:

- **Malicious contracts** with re-entrancy or callback exploits
- **Spam NFTs** with no real value (airdrop attacks)
- **Non-standard tokens** that don't properly implement ERC-721/1155

> **Security:** The NFTVault uses OpenZeppelin's `ERC721Holder` and `ERC1155Holder` safe receiver interfaces. All transfers use `safeTransferFrom` to ensure the vault can properly receive and return tokens. Reentrancy guards protect all withdrawal paths.

NFT BETTING + P2P INTEGRATION

NFT bets can be combined with P2P orders by setting `isNFT = true` in the order struct. This enables sophisticated betting scenarios:

SCENARIO	DESCRIPTION	COLLATERAL
NFT vs Tokens	One side stakes an NFT, the other stakes SBET tokens	NFT + Tokens (asymmetric)
NFT vs NFT	Both sides stake NFTs — winner takes both	NFT + NFT (symmetric)
NFT + Token vs Tokens	One side enhances their NFT stake with additional tokens	Mixed collateral

NFT VALUATION & SETTLEMENT

When NFTs are used as betting collateral, the protocol handles valuation at the order level — makers and takers agree on terms before filling. This peer-negotiated model avoids the complexity and manipulation risk of on-chain NFT price oracles.

- **No price oracle dependency:** The agreed trade terms implicitly value the NFT
- **Winner takes all:** The winning side receives both the NFT(s) and any token collateral
- **Fee basis:** Fees are calculated on the token component only — no fee on NFT transfers

CHAPTER 8

8. Prediction Markets

SBET's prediction market engine uses the Logarithmic Market Scoring Rule (LMSR) to provide automated market making for multi-outcome events. Unlike P2P trading where prices are set by makers, prediction markets have an algorithmic market maker that continuously quotes prices for all outcomes.

LMSR ALGORITHM

The LMSR, developed by Robin Hanson, is the gold standard for prediction market pricing. It guarantees bounded loss for the market maker while providing unlimited liquidity:

$$\text{Cost Function: } C(q) = b \cdot \ln(\Sigma\ e^{q_i/b})$$

$$\text{Price of outcome i: } p_i = e^{q_i/b} \div \Sigma\ e^{q_j/b}$$

$$\text{Where: } q_i = \text{outstanding shares of outcome i, } b = \text{liquidity parameter}$$

The liquidity parameter `b` controls the market maker's price sensitivity. Higher `b` means more liquidity (less price impact per trade), but higher maximum loss for the market maker subsidy.

ARCHITECTURE

PREDICTIONMARKET.SOL

Market lifecycle management: creation, trading, resolution, and payout. Manages market state, outcome counts, and resolution logic including M-of-N dispute mechanism.

PREDICTIONAMM.SOL

LMSR cost function implementation. Calculates buy/sell costs, manages share balances, and handles the automated pricing algorithm.

SHARE TRADING

Users interact with prediction markets by buying and selling outcome shares:

ACTION	EFFECT	COST
Buy shares	Increases position in outcome i	$C(q + \Delta q) - C(q)$ — always positive, increases with quantity
Sell shares	Decreases position in outcome i	$C(q) - C(q - \Delta q)$ — refund, decreases with quantity sold
Redeem (win)	Convert winning shares to tokens	1 token per share of winning outcome
Redeem (lose)	Losing shares expire worthless	0 tokens returned

M-OF-N DISPUTE RESOLUTION

Prediction markets use a multi-oracle dispute mechanism for outcome resolution:

1. **Initial resolution:** The designated oracle submits the outcome

2. **Challenge period:** Anyone can challenge the outcome by posting a dispute bond
3. **Multi-oracle vote:** If challenged, M-of-N designated resolvers vote on the correct outcome
4. **Finalization:** The majority-voted outcome becomes final; losing challengers forfeit their bond

Why LMSR? Unlike order-book or AMM-based prediction markets, LMSR guarantees liquidity at all price points. A market maker's maximum loss is bounded by `b · ln(n)` where n is the number of outcomes. This means the protocol can provide guaranteed liquidity with a known, bounded subsidy cost.

LMSR WORKED EXAMPLE

Example: 3-Outcome Market (Win/Draw/Lose)

Initial state: $b = 100$, $q = [0, 0, 0]$ (no shares sold)
Initial prices: $p = [0.333, 0.333, 0.333]$ (equal probability)

User buys 10 "Win" shares:
$\text{Cost} = 100 \cdot \ln(e^{10/100} + e^{0/100} + e^{0/100}) - 100 \cdot \ln(e^{0/100} + e^{0/100} + e^{0/100})$
Cost ≈ 3.44 tokens

New prices: p_win ≈ 0.366, p_draw ≈ 0.317, p_lose ≈ 0.317
Effect: Buying "Win" shares pushed its price up and the others down — exactly as expected.

MARKET LIFECYCLE

1. Market Created → 2. Trading Open → 3. Event Occurs → 4. Oracle Resolves → 5. Challenge Period → 6. Final Payout

USE CASES FOR PREDICTION MARKETS

CATEGORY	EXAMPLE MARKETS	OUTCOMES
Tournament Winners	World Cup winner, Champions League winner	8-32 teams
Season Awards	MVP, Golden Boot, Ballon d'Or	5-20 candidates
Transfer Markets	Will Player X transfer to Club Y before deadline?	2-4 outcomes
League Standings	Which team finishes top 4? Relegation predictions	2-20 teams
Record Breakers	Will a player score 30+ goals this season?	Binary (Yes/No)

PREDICTION MARKETS VS POOL BETTING

FEATURE	PREDICTION MARKETS (LMSR)	POOL BETTING (PARI-MUTUEL)
Pricing	Continuous algorithmic pricing	Final odds at pool close
Liquidity	Always available (bounded loss)	Depends on participant count

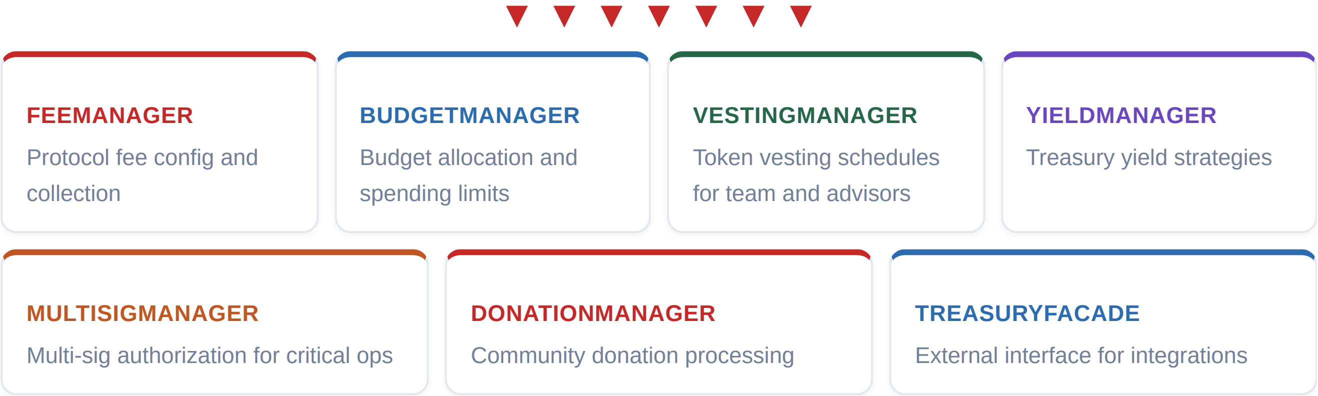
Trading	Buy and sell shares anytime	Stake only (no exit before settlement)
Best for	Information discovery, complex events	Simple, popular sporting events
Resolution	Multi-oracle with dispute mechanism	Single oracle finalization

CHAPTER 9

9. Treasury System

The SBET treasury is a hub-and-spoke smart contract system that manages all protocol funds with defense-in-depth security. It consists of a central `Treasury.sol` hub contract and seven specialized module contracts, each handling a specific financial operation with independent access controls and daily limits.

HUB-AND-SPOKE ARCHITECTURE



FEEMANAGER
Protocol fee config and collection

BUDGETMANAGER
Budget allocation and spending limits

VESTINGMANAGER
Token vesting schedules for team and advisors

YIELDMANAGER
Treasury yield strategies

MULTISIGMANAGER
Multi-sig authorization for critical ops

DONATIONMANAGER
Community donation processing

TREASURYFACADE
External interface for integrations

TREASURY MODULES

MODULE	RESPONSIBILITY	ACCESS CONTROL
FeeManager	Configures fee rates for all protocol operations. Collects fees from P2P trades, pool bets, NFT stakes, and prediction market trades.	Treasury admin only
BudgetManager	Manages spending budgets with daily limits. Each budget category (development, marketing, operations) has independent caps.	Multisig required above threshold
VestingManager	Creates and manages token vesting schedules. Supports cliff periods, linear vesting, and revocable grants.	Treasury admin + timelock
YieldManager	Deploys idle treasury funds into approved yield strategies. Tracks positions and manages withdrawals.	Multisig + whitelist
MultisigManager	Provides M-of-N signature requirements for high-value operations. Manages signer sets and threshold changes.	Self-governing (signers manage signers)
DonationManager	Processes community donations to the protocol treasury. Tracks donor history and issues receipts.	Public (anyone can donate)
TreasuryFacade	Read-only external interface. Provides a clean API for integrators and dashboards to query treasury state.	Public (read-only)

SECURITY CONTROLS

DAILY LIMITS

Every module enforces daily spending limits. Exceeding the limit requires multisig approval, preventing single-actor fund drainage.

EMERGENCY PAUSE

The treasury can be paused independently of the betting contracts. Pause halts all outflows while still accepting inflows (fees).

MIGRATION TIMELOCK

Treasury migration to a new contract requires a timelock period, giving the community time to review and object before funds move.

ROLE-BASED ACCESS

OpenZeppelin `AccessControl` with granular roles: ADMIN, OPERATOR, MANAGER, VIEWER. Least-privilege enforced.

FUND FLOW

Betting fees collected → **Treasury Hub receives** → **60% to staker rewards** | **30% to treasury operations** | **10% burned**

TOKEN METRICS TRACKING

The treasury system tracks comprehensive on-chain metrics for transparency and auditing:

```
struct TokenMetrics {
    uint256 totalDeposited;    // Cumulative inflows
    uint256 totalWithdrawn;    // Cumulative outflows
    uint256 totalFees;         // Cumulative fees collected
    uint256 lastUpdated;       // Block timestamp of last update
}
```

These metrics are queryable via the `TreasuryFacade` contract, enabling any external dashboard or analytics platform to display real-time treasury health.

VESTING SCHEDULES

The `VestingManager` supports multiple vesting configurations for different stakeholder categories:

STAKEHOLDER	CLIFF	VESTING PERIOD	TYPE	REVOCABLE
Core Team	12 months	36 months (linear)	Linear vesting	No
Advisors	6 months	24 months (linear)	Linear vesting	Yes (by multisig)
Ecosystem Grants	None	12-24 months	Milestone-based	Yes (by multisig)
Strategic Partners	3 months	18 months (linear)	Linear vesting	Yes (by multisig)

BUDGET ALLOCATION

The 30% of protocol fees that flow to treasury operations are allocated across categories:

BUDGET CATEGORY	ALLOCATION	DAILY LIMIT	PURPOSE
Development	40%	Configurable	Smart contract development, frontend, infrastructure
Marketing	25%	Configurable	Community growth, partnerships, listings
Operations	20%	Configurable	Oracle costs, gas subsidies, infrastructure
Reserve	15%	Multisig only	Emergency fund, strategic opportunities

CHAPTER 10

10. Integrator Hub

The Integrator Hub (`IntegratorHub.sol`) enables third-party platforms to build on top of SBET Protocol with built-in revenue sharing. Any developer or platform can register as an integrator and earn a share of the fees generated through their front-end.

SELF-SERVICE REGISTRATION

Integrator onboarding is fully permissionless:

1. **Register:** Call `register()` with integrator metadata (name, website, payout address)
2. **Receive ID:** The hub assigns a unique integrator ID used to attribute trades
3. **Embed ID:** Include the integrator ID when submitting orders through SBET Core
4. **Earn fees:** A configured percentage of each attributed trade's protocol fee flows to the integrator

FEE SHARING MODEL

FEE COMPONENT	RECIPIENT	DESCRIPTION
Base protocol fee	SBET Treasury	Core protocol fee from every trade
Integrator share	Integrator wallet	Percentage of protocol fee credited to integrator's balance
Remainder	Treasury → stakers/burn	Standard treasury distribution (staker rewards + burn)

PAYOUT SWEEPS

Integrator earnings accumulate in the hub contract and can be swept to the integrator's wallet at any time. Batch sweep operations allow multiple integrators to claim in a single transaction, reducing gas costs for the protocol operator.

USE CASES

SPORTS BETTING APPS

Build custom sports betting UIs that connect to SBET's liquidity and earn fees on every bet placed through the app.

TELEGRAM / DISCORD BOTS

Create social betting bots that let users place bets directly in chat, with the bot earning integrator fees.

AGGREGATOR PLATFORMS

Odds aggregators can route orders to SBET when it offers the best price, earning fees for order flow.

DEFI PROTOCOLS

Lending, yield, and portfolio protocols can integrate SBET positions as collateral or yield sources.

Developer Incentive: The integrator model aligns incentives — the more volume an integrator drives, the more they earn. This creates a network of motivated distribution partners building the SBET ecosystem without any centralized business development effort.

INTEGRATOR REGISTRATION FLOW

1. Call register() → 2. Receive Integrator ID → 3. Embed ID in order submissions → 4. Fees accumulate → 5. Sweep to wallet

INTEGRATOR DASHBOARD

Each integrator can query their performance metrics via the `IntegratorHub` contract:

METRIC	DESCRIPTION	QUERY METHOD
Total volume routed	Cumulative value of all orders attributed to this integrator	On-chain read
Fees earned	Total integrator fee share accumulated	On-chain read
Pending balance	Unswept fee balance available for withdrawal	On-chain read
Order count	Number of orders attributed to this integrator	Event-based indexing
Active users	Unique wallets that have traded through this integrator	Event-based indexing

INTEGRATOR TIERS

As integrators grow their volume, they unlock higher fee shares:

TIER	VOLUME THRESHOLD	FEE SHARE	BENEFITS
STARTER	0+	Base rate	Basic integration, standard API access
GROWTH	Volume milestone 1	Enhanced rate	Priority support, analytics dashboard
PRO	Volume milestone 2	Premium rate	Custom fee structures, co-marketing, early access to new features

TECHNICAL INTEGRATION GUIDE

Integrating with SBET requires minimal development effort:

```
// 1. Register as integrator (one-time)
const tx = await integratorHub.register("MyApp", "https://myapp.com", payoutAddress);
const integratorId = /* extract from event */;

// 2. Include integrator ID when submitting orders
const order = {
  ...standardOrder,
  // Integrator ID is passed as a parameter to the fill function
};
await sbetCore.fillOrder(order, signature, integratorId);

// 3. Sweep accumulated fees
await integratorHub.sweep(integratorId);
```

CHAPTER 11

11. AI-Powered Analytics

SBET Protocol integrates a dual-provider AI system that provides real-time market analysis, confidence scoring, and intelligent suggestions to users. The AI layer is strictly read-only — it never signs transactions or controls funds.

DUAL-PROVIDER ARCHITECTURE

CLAUDE (ANTHROPIC)

Advanced reasoning model for complex market analysis, multi-factor assessments, and nuanced betting strategy recommendations.

GPT-4O (OPENAI)

High-speed model for real-time odds analysis, quick market scans, and conversational interaction.

Users can switch between providers at runtime based on their preference. API keys are stored in **sessionStorage only** (tab-scoped, cleared on browser close) and are never sent to any SBET backend.

KEY FEATURES

Confidence Scoring

The AI system generates per-market confidence scores (0–100%) by analyzing:

- Current odds and implied probabilities across all available markets
- Odds movement patterns (sharp money indicators)
- Live match statistics (possession, shots, momentum) for in-play events
- Historical matchup data and form analysis
- Liquidity depth and market efficiency metrics

AI Suggestions

The system produces structured suggestions with:

```
interface AISuggestion {
  marketId: string;           // Target market
  pick: string;               // Recommended selection
  odds: number;               // Current odds
  confidence: number;         // 0–1 confidence score
  reason: string;             // Human-readable rationale
  kellyFraction: number;      // Kelly criterion fraction
  recommendedStake: number;   // Suggested stake size
}
```

Simulated Market Maker

The AI can simulate a market maker that continuously monitors markets and suggests liquidity actions:

ACTION	TRIGGER	EFFECT
`add_liquidity`	Market spread too wide or liquidity too thin	Suggests adding liquidity at fair-value prices
`remove_liquidity`	Risk exceeds thresholds or event approaching	Suggests pulling liquidity from risky positions

`adjust_odds`	New information changes fair value	Suggests odds adjustment with reasoning

SECURITY MODEL

Critical Design Decision: The AI layer is completely isolated from transaction signing and fund control. It operates as an advisory-only system:

- API keys in sessionStorage only — never persisted to disk or sent to SBET servers
- No access to wallet private keys or signing capabilities
- All suggestions are informational — users must manually approve any transaction
- No ability to modify smart contract state or parameters

MARKET CONTEXT

The AI system receives a structured market context snapshot containing current matches, odds, live scores, wallet state, and open bet slip items. This context is sent directly to the AI provider from the user's browser — SBET servers never see the data.

CONTEXT DATA STRUCTURE

The market context passed to the AI provider includes:

```
interface MarketContext {
  matches: Array<{
    id: string;
    lg: string;                // League
    home: string;              // Home team name
    away: string;              // Away team name
    status: string;            // "live", "upcoming", "finished"
    homeOdds: number;
    drawOdds?: number;
    awayOdds: number;
    homeLiq?: number;          // Home side liquidity
    awayLiq?: number;          // Away side liquidity
    minute?: number;           // Current minute (live only)
    homeScore?: number;
    awayScore?: number;
  }>;
  walletConnected: boolean;
  walletBalance: number;
  openSlip: Array<{ pick: string; odds: number; stake: string }>;
  networkName: string;
}
```

KELLY CRITERION INTEGRATION

The AI system uses the Kelly Criterion for optimal bet sizing recommendations. The Kelly formula determines the fraction of bankroll to wager based on the edge and odds:

```
f* = (bp − q) ÷ b

Where: f* = optimal fraction, b = net odds (decimal − 1),
p = estimated win probability, q = 1 − p
```

The AI applies a fractional Kelly approach (typically 25-50% of full Kelly) to reduce variance and account for estimation uncertainty. This conservative approach protects users from overbetting on high-confidence but potentially miscalibrated signals.

AI ETHICS & RESPONSIBLE USE

NO GUARANTEED OUTCOMES

AI suggestions are probabilistic assessments, not guarantees. The system explicitly disclaims prediction accuracy and encourages users to do their own research.

BANKROLL PROTECTION

Kelly criterion sizing ensures recommended stakes are proportional to edge, preventing users from risking excessive amounts on single events.

TRANSPARENCY

Every suggestion includes a human-readable rationale explaining why the AI reached its conclusion, allowing users to evaluate the reasoning.

PRIVACY

All AI processing happens between the user's browser and the AI provider. No betting data, wallet balances, or analysis results touch SBET servers.

CHAPTER 12

12. Tokenomics

SBET is an ERC-20 token on Ethereum mainnet with a fixed, immutable supply of 3,500,000,000 (3.5 billion) tokens. There is no minting function — the total supply was set at deployment and can never increase.

TOKEN DETAILS

PROPERTY	VALUE
Name	Sports BET
Symbol	SBET
Standard	ERC-20
Chain	Ethereum Mainnet
Contract	`0x2ed2cc2c858a8a8219fd2f2d9e170285dbd02756`
Total Supply	3,500,000,000 SBET
Decimals	18
Mintable	No — fixed supply, no mint function

TOKEN ALLOCATION

76%	**10%**	**8%**	**6%**
PUBLIC SALE / DEPLOYER	TEAM (LOCKED)	ICO	PRESALE / MARKETING

CATEGORY	ALLOCATION	TOKENS	DETAILS
Public Sale / Deployer	76%	2,660,000,000	Largest allocation ensures broad distribution and deep liquidity from day one
Team (Locked)	10%	350,000,000	Vesting schedules maintain long-term alignment with the community
ICO	8%	280,000,000	Initial coin offering allocation for early supporters
Presale	3%	105,000,000	Early-stage presale participants
Marketing, Dev & Bounty	3%	105,000,000	Community growth, developer grants, bug bounties, partnerships

DEFLATIONARY MECHANICS

SBET implements a continuous burn mechanism that reduces the circulating supply over time:

FEE BURN (10%)

10% of all protocol fees are permanently burned. As trading volume grows, the burn rate accelerates, creating increasing scarcity.

NO INFLATION

With no minting capability, every burned token permanently reduces the total supply. The token is strictly deflationary.

```
Circulating Supply = 3,500,000,000 − Σ(Burned Tokens) − Locked Tokens

Burn Rate = Total Protocol Fees × 10%
```

TOKEN UTILITY

UTILITY	DESCRIPTION
Betting Collateral	Primary token for P2P orders, pool stakes, and prediction market shares
Staking Rewards	Stake SBET to earn 60% of protocol fees
Governance	Voting weight in protocol governance decisions (planned DAO transition)
Fee Discounts	Betting with SBET incurs only 2% fee vs 3.5% for other tokens — a 43% discount
Integrator Bonds	Integrators stake SBET as a good-faith bond for premium features

> **Value Accrual:** SBET's tokenomics create a virtuous cycle — more betting volume generates more fees, which drives more staking rewards (attracting holders), more token burns (increasing scarcity), and more integrator interest (driving more volume). This flywheel effect means the token's utility grows with protocol adoption.

VALUE ACCRUAL FLYWHEEL

More betting volume → More protocol fees → **Higher staking rewards** → More tokens staked → Reduced circulating supply

Fee burns (10%) → **Permanent supply reduction** → Increased scarcity → Attracts new holders → Integrator growth

TOKEN FLOW DIAGRAM

▼

BETTING COLLATERAL

Users deposit SBET as collateral for P2P trades, pool stakes, and prediction market shares. Tokens are escrowed during active bets.

FEE COLLECTION

Protocol fees (2% for SBET token, 3.5% for other tokens) are collected on each operation and routed to the Treasury Hub for distribution.

REWARD DISTRIBUTION

60% of fees → stakers (proportional), 30% → treasury operations, 10% → permanent burn.

SUPPLY SCHEDULE

With no minting and continuous burning, SBET's effective supply decreases over time:

YEAR	MAX SUPPLY	ESTIMATED BURN	EFFECTIVE MAX SUPPLY
2022 (Launch)	3,500,000,000	—	3,500,000,000
2026	3,500,000,000	Growing with volume	Decreasing
2027+	3,500,000,000	Accelerating with multi-chain	Continuously decreasing

Deflationary Model: Unlike inflationary protocols that dilute holders via token emissions, SBET's fixed supply + burn model ensures every token holder's percentage ownership can only increase over time (assuming no selling). This aligns long-term holder incentives with protocol growth.

CHAPTER 13

13. Fee Structure

SBET Protocol's fees are fully transparent and enforced on-chain. Unlike traditional sportsbooks that embed hidden margins (vig) into odds, SBET charges explicit, predictable fees that users can verify before placing any bet.

SBET TOKEN FEE ADVANTAGE

The protocol incentivizes use of the native SBET token by offering a significantly lower fee rate compared to other ERC-20 tokens or ETH:

2%	3.5%	43%
FEE WITH SBET TOKEN	FEE WITH OTHER TOKENS	SAVINGS WITH SBET

Why the discount? By using SBET as the betting collateral, users directly contribute to the token's velocity and utility. The 2% fee for SBET holders represents a 43% discount over the 3.5% standard rate — a powerful incentive that drives organic token demand while keeping fees lower than any competitor.

FEE SCHEDULE

OPERATION	SBET TOKEN FEE	OTHER TOKEN FEE	DESCRIPTION
P2P TRADING			
Order fill (maker)	**1%**	1.75%	Fee on maker's collateral at fill
Order fill (taker)	**1%**	1.75%	Fee on taker's collateral at fill
Order cancellation	Free		Off-chain cancellation (nonce invalidation)
POOL BETTING			
Pool payout	**2%**	3.5%	Deducted from pool payout at settlement
Pool creation	Free		No fee to create a pool
NFT BETTING			
NFT stake	**2%**	3.5%	Per-collection configurable rate (base rate shown)
NFT escrow/release	Free		No fee for vault operations
PREDICTION MARKETS			
Share purchase	**2%**	3.5%	Applied to LMSR cost at purchase
Share sale	**2%**	3.5%	Applied to LMSR refund at sale
Share redemption	Free		Winning shares redeemed at par value

CASH BACK / WITHDRAWAL		
Withdrawal	Free (gas only)	No protocol fee — users only pay network gas

FEE DISTRIBUTION

60%
STAKER REWARDS

30%
TREASURY OPERATIONS

10%
TOKEN BURN

COMPARISON WITH TRADITIONAL SPORTSBOOKS

PLATFORM TYPE	EFFECTIVE FEE	TRANSPARENCY	FEE DESTINATION
Traditional sportsbook	5-15% (vig)	Hidden in odds	Operator profit
Crypto sportsbook (CeFi)	3-8%	Partially visible	Operator profit
DeFi betting (AMM)	2-5%	On-chain visible	LPs + protocol
SBET Protocol (SBET token)	**2%**	**Fully on-chain**	**Stakers (60%) + burn (10%)**
SBET Protocol (other tokens)	3.5%	Fully on-chain	Stakers (60%) + burn (10%)

> **Fee Advantage:** SBET's peer-to-peer model eliminates the house edge entirely. The protocol charges just 2% when betting with the SBET token (or 3.5% with other tokens) — 3-7x lower than traditional sportsbooks. Moreover, 70% of fees flow back to the token ecosystem (staker rewards + burn), making SBET users the primary beneficiaries. Free cash back and zero withdrawal fees complete the low-cost picture.

FEE ECONOMICS DEEP DIVE

Understanding SBET's fee advantage requires comparing the total cost of placing a bet across different platform types:

Traditional Sportsbook Fee Breakdown

COST COMPONENT	TRADITIONAL SPORTSBOOK	SBET PROTOCOL
Odds margin (vig)	5-15%	0% (peer-to-peer pricing)
Platform fee	0% (embedded in vig)	2% with SBET / 3.5% with other tokens
Deposit fee	0-3% (card/wire)	Gas only (~$0.01 on L2)
Withdrawal fee	$0-25 + 1-3 day delay	Free (gas only), instant
Currency conversion	1-3%	0% (native crypto)
Total effective cost	**6-24%**	**2-3.5%**

Fee Revenue Model

SBET's fee model creates sustainable revenue without extracting excessive value from users:

```
Annual Fee Revenue = Daily Betting Volume × Average Fee Rate × 365

Staker Annual Yield = (Annual Fee Revenue × 60%) ÷ Total Staked Value
```

As the protocol scales across multiple chains and integrator partners, fee revenue grows proportionally — creating a direct link between ecosystem growth and token holder returns.

FEE GOVERNANCE

Fee rates are governable parameters that can be adjusted through the governance process:

- **Current:** Multisig can adjust fee rates within defined bounds
- **Phase 2:** Token holder advisory vote on fee changes
- **Phase 3:** Full DAO governance of fee parameters

Fee changes are bounded by smart contract-enforced minimums and maximums to prevent governance attacks that could either extract excessive fees or eliminate protocol revenue entirely.

CHAPTER 14

14. Staking & Rewards

SBET's staking system allows token holders to earn a share of all protocol fees by locking their tokens. Stakers collectively receive 60% of all fees generated across every betting primitive — P2P trading, pool betting, NFT staking, and prediction markets.

REVENUE SHARING MODEL

60%	30%	10%
TO STAKERS	TO TREASURY	BURNED

HOW STAKING WORKS

1. **Stake:** Users deposit SBET tokens into the staking contract
2. **Earn:** Revenue from protocol fees accrues to stakers proportionally by share of total stake
3. **Claim:** Rewards can be claimed at any time without unstaking
4. **Unstake:** Users can withdraw their staked tokens (subject to any cooldown period)

```
User Reward = (User Stake ÷ Total Staked) × Protocol Fees × 60%
```

REWARD SOURCES

SOURCE	FEE RATE (SBET / OTHER)	STAKER SHARE (60%)	VOLUME SENSITIVITY
P2P Trading	2% / 3.5%	1.2% / 2.1%	High — grows with trading volume
Pool Betting	2% / 3.5%	1.2% / 2.1%	Medium — grows with event popularity
NFT Betting	2% / 3.5%	1.2% / 2.1%	Medium — tied to NFT market activity
Prediction Markets	2% / 3.5%	1.2% / 2.1%	High — grows with prediction market usage

STAKING BENEFITS

PASSIVE INCOME

Earn a proportional share of all protocol fees without actively betting. Revenue is generated 24/7 from global sports events.

GOVERNANCE WEIGHT

Staked tokens count toward governance voting power. Longer staking duration may provide boosted voting weight in future DAO governance.

FEE DISCOUNTS

Stakers receive reduced protocol fees on their own trades, creating an incentive for active participants to also stake.

DEFLATIONARY PRESSURE

Staking locks tokens, reducing circulating supply. Combined with the 10% burn, this creates sustained deflationary pressure.

Yield Projection: As protocol volume grows, staking yields increase proportionally. With SBET's multi-primitive approach covering P2P trading, pool betting, NFT staking, and prediction markets, fee revenue is diversified across multiple sources — reducing dependency on any single market or event.

STAKING MECHANICS

The staking system is designed for simplicity and security:

MECHANISM	IMPLEMENTATION	USER EXPERIENCE
Deposit	ERC-20 transfer to staking contract	Single transaction, instant confirmation
Reward accrual	Pro-rata share of fee distribution events	Rewards accumulate automatically, no claim needed to accrue
Reward claim	Pull-based claim function	Claim anytime without unstaking principal
Withdrawal	Unstake principal + unclaimed rewards	Full withdrawal in one transaction
Compounding	Claim + restake in separate txs	Manual compounding (auto-compound planned)

STAKING SECURITY

NO IMPERMANENT LOSS

Unlike LP staking, SBET staking has no impermanent loss risk. Users deposit SBET and always withdraw SBET — no token pair exposure.

NON-CUSTODIAL

Staked tokens remain in the smart contract (not in any wallet). Only the depositor can withdraw. No admin withdrawal capability.

REVENUE DIVERSIFICATION

SBET staking yields are uniquely diversified because they draw from four independent betting primitives:

P2P TRADING FEES

Generated from order fills. Volume correlates with market liquidity and trading activity. Highest during major sporting events.

POOL BETTING FEES

Generated from pool settlements. Steady revenue from popular events with high participation. Seasonal patterns (football, basketball, etc.).

NFT BETTING FEES

Generated from NFT stakes. Correlated with NFT market activity and sports collectible trends.

PREDICTION MARKET FEES

Generated from share trading. Active during long-running events (seasons, tournaments, transfers).

CHAPTER 15

15. Governance

SBET Protocol is transitioning from a multisig-controlled governance model to a fully decentralized DAO. This progressive decentralization ensures security during the early protocol phase while building toward community-owned governance.

CURRENT MODEL: MULTISIG GOVERNANCE

The protocol is currently governed by a multisig arrangement managed through the `MultisigManager.sol` contract:

PARAMETER	CURRENT SETTING
Governance model	Multisig (M-of-N signatures required)
Controlled parameters	Fee rates, oracle addresses, emergency pause, treasury allocations
Timelock	Required for treasury migration and critical parameter changes
Emergency powers	Time-bounded pause (auto-resumes after window expires)

PLANNED: FULL DAO TRANSITION

Phase 1: Multisig → Phase 2: Multisig + Token Vote → Phase 3: Full DAO → Phase 4: Immutable

PHASE	GOVERNANCE	SCOPE	TIMELINE
PHASE 1	Multisig only	Full parameter control	Current
PHASE 2	Multisig + advisory token vote	Token holders vote on proposals; multisig executes	H2 2026
PHASE 3	Full DAO governance	Token vote controls all parameters; timelock execution	2027
PHASE 4	Immutable core + DAO periphery	Core contracts made immutable; DAO governs periphery only	2028+

GOVERNABLE PARAMETERS

FEE PARAMETERS

Protocol fee rates for each betting primitive, integrator fee share percentages, NFT per-collection fee rates.

ORACLE CONFIGURATION

Approved oracle addresses for match finalization, prediction market resolvers, dispute bond amounts.

TREASURY CONTROLS

Daily spending limits, budget allocations, yield strategy whitelists, vesting schedule modifications.

SYSTEM OPERATIONS

Emergency pause/unpause, contract upgrades, NFT collection blacklists, integrator registration rules.

GOVERNANCE PARAMETER DETAILS

PARAMETER	CURRENT CONTROL	RANGE/CONSTRAINTS	CHANGE PROCESS
P2P fee rate	Multisig	0-5% (basis points)	Proposal → multisig vote → immediate effect
Pool fee rate	Multisig	0-10% (basis points)	Proposal → multisig vote → immediate effect
Oracle address	Multisig	Valid Ethereum address	Proposal → multisig vote → timelock → effect
Daily spending limit	Multisig	Min floor enforced	Proposal → multisig vote → immediate effect
Emergency pause	Any signer	Boolean (on/off)	Single signer can pause; multisig required to unpause
Treasury migration	Full multisig	Valid contract address	Proposal → full multisig → timelock → migration
NFT blacklist	Multisig	Collection address	Proposal → multisig vote → immediate effect
LMSR liquidity (b)	Multisig	Per-market configurable	Set at market creation; cannot change after trading starts

DAO TRANSITION PLAN

The transition from multisig to full DAO governance is designed to be gradual and safe:

PHASE 2: ADVISORY VOTING

Token holders can vote on proposals via on-chain governance contracts. Votes are advisory — the multisig retains execution power but commits to following majority votes on non-critical parameters.

PHASE 3: BINDING VOTING

Token votes directly control parameter changes via on-chain execution. Timelock on all changes. Multisig retains emergency-only powers (pause). Quorum and supermajority requirements prevent governance attacks.

PHASE 4: IMMUTABILITY

Core contracts are made permanently immutable — no governance can modify them. Only periphery contracts (fee rates, oracle lists) remain governable. The protocol becomes truly trustless.

ATTACK RESISTANCE

Governance attack vectors (flash loan voting, minority takeover) are mitigated via: voting escrow (time-locked staking), quorum requirements, proposal thresholds, and execution timelocks.

CHAPTER 16

16. Security & Audit

SBET Protocol underwent a comprehensive security audit conducted by **Versus Security** in February 2026, covering all 24 smart contracts, 25,000+ lines of Solidity code, and 883 test cases. The audit identified 146 findings across all severity levels, all of which have been addressed.

AUDIT OVERVIEW

146	24	25K+	883
TOTAL FINDINGS	CONTRACTS REVIEWED	LINES ANALYZED	TEST CASES

FINDINGS BY SEVERITY

SEVERITY	COUNT	STATUS	DESCRIPTION
CRITICAL	2	FIXED	EIP-712 replay, reentrancy in settlement — both patched immediately
HIGH	8	FIXED	Access control, integer overflow, oracle manipulation vectors
MEDIUM	22	FIXED	Gas optimization, edge cases in pool finalization, nonce gaps
LOW	45	FIXED	Code quality, documentation, naming conventions
INFORMATIONAL	69	ACKNOWLEDGED	Best practice suggestions, gas hints, style recommendations

AUDIT METHODOLOGY

MANUAL REVIEW

Line-by-line code review by senior Solidity security researchers. Focus on business logic, access control, and economic attacks.

STATIC ANALYSIS

Slither automated analysis for common vulnerability patterns: reentrancy, unchecked returns, storage collisions, shadowing.

INVARIANT TESTING

Echidna fuzzer with property-based tests. Verified economic invariants: total supply conservation, payout bounds, fee consistency.

ECONOMIC MODELING

Attack simulations for front-running, sandwich attacks, oracle manipulation, and flash loan exploitation scenarios.

CRITICAL FIXES APPLIED

FINDING	RISK	FIX

EIP-712 domain replay across chains	CRITICAL	Added keccak256 salt + dynamic chainId verification to domain separator
Settlement reentrancy in pool payout	CRITICAL	Applied CEI pattern + OpenZeppelin ReentrancyGuard to all payout functions
Missing access control on oracle setter	HIGH	Added role-based access control with multisig requirement
Integer overflow in fee calculation	HIGH	Solidity 0.8.x checked arithmetic + explicit bounds validation

Full Audit Report: The complete Versus Security audit report with all 146 findings, descriptions, and remediation details is available at **sbettoken.org/audit**

AUDIT SCOPE

The Versus Security audit covered every contract in the SBET protocol, including:

DOMAIN	CONTRACTS REVIEWED	FOCUS AREAS
Core Engine	SBET.sol, SBETQuery.sol	EIP-712 verification, order matching, pool logic, match finalization
NFT System	NFTVault, NFTManager, NFTFeeManager	Escrow safety, ERC-721/1155 callbacks, fee calculation
Prediction Markets	PredictionMarket, PredictionAMM	LMSR implementation, share accounting, dispute resolution
Treasury	Treasury, Facade, Fee/Budget/Vesting/Yield/Multisig/Donation Managers	Access control, daily limits, fund isolation, migration safety
Integration	IntegratorHub	Registration, fee attribution, payout sweep

TEST COVERAGE

The protocol's 883 test cases cover critical paths across all contract domains:

UNIT TESTS

Individual function testing for all public and external functions. Edge cases for zero values, max values, boundary conditions, and invalid inputs.

INTEGRATION TESTS

End-to-end flows: order creation → fill → settlement → claim. Pool creation → stake → finalize → payout. NFT stake → match → release.

INVARIANT TESTS (FUZZING)

Echidna property-based tests verify: total supply conservation, payout ≤ pool total, fees ≥ 0, nonce monotonicity, access control enforcement.

ATTACK SIMULATIONS

Tests for: reentrancy, front-running, sandwich attacks, flash loan exploitation, EIP-712 replay, nonce manipulation, oracle spoofing.

ONGOING SECURITY PRACTICES

- **Continuous monitoring:** Smart contract events monitored for anomalous behavior patterns
- **Slither CI/CD:** Static analysis runs on every commit to catch regressions
- **Dependency auditing:** Regular `npm audit` and OpenZeppelin version tracking
- **Second audit planned:** A follow-up audit is planned before protocol ossification (Phase 4)
- **Bug bounty:** Responsible disclosure program planned for launch

CHAPTER 17

17. Smart Contract Inventory

SBET Protocol consists of 24 production smart contracts organized across four functional domains. Each contract is independently audited, tested, and deployed.

COMPLETE CONTRACT LIST

#	CONTRACT	DOMAIN	DESCRIPTION
CORE & ENTRY (2 CONTRACTS)			
1	`SBET.sol`	CORE	Main entry point. EIP-712 domain, P2P order matching, pool betting, match management, nonce tracking.
2	`SBETQuery.sol`	CORE	Read-only query aggregator. Provides batch data fetching for frontend (matches, pools, orders, balances).
NFT SYSTEM (3 CONTRACTS)			
3	`NFTVault.sol`	NFT	Secure ERC-721/1155 escrow. Implements safe receiver interfaces. Only NFTManager can withdraw.
4	`NFTManager.sol`	NFT	NFT bet coordination. Manages stake records, matching, and settlement.
5	`NFTFeeManager.sol`	NFT	Per-collection fee rates and blacklist management.
PREDICTION MARKETS (2 CONTRACTS)			
6	`PredictionMarket.sol`	PREDICTION	Market lifecycle (create, trade, resolve, payout). M-of-N dispute resolution. Multi-outcome support.
7	`PredictionAMM.sol`	PREDICTION	LMSR cost function. Automated pricing, share issuance, and buy/sell cost calculation.
TREASURY SYSTEM (7 CONTRACTS)			
8	`Treasury.sol`	TREASURY	Central hub. Role-based access, emergency pause, fund custody, module coordination.
9	`TreasuryFacade.sol`	TREASURY	External read-only interface for integrators and dashboards.
10	`FeeManager.sol`	TREASURY	Protocol-wide fee configuration and collection routing.
11	`BudgetManager.sol`	TREASURY	Spending budgets with daily limits and category-based allocation.
12	`VestingManager.sol`	TREASURY	Token vesting with cliff, linear, and revocable schedules.
13	`YieldManager.sol`	TREASURY	Treasury yield deployment into whitelisted strategies.
14	`MultisigManager.sol`	TREASURY	M-of-N authorization for critical treasury operations.
15	`DonationManager.sol`	TREASURY	Community donation processing and donor tracking.

INTEGRATION (1 CONTRACT)			
16	`IntegratorHub.sol`	INTEGRATION	Self-service partner registration, fee sharing, and payout sweeps.
TOKEN (1 CONTRACT)			
17	`SBETToken.sol`	TOKEN	ERC-20 token (3.5B fixed supply, no minting, deployed on Ethereum mainnet).
LIBRARIES & HELPERS (7 CONTRACTS)			
18-24	`Various`	LIBRARY	Shared libraries for math operations, validation helpers, type conversions, event definitions, error definitions, constants, and storage layout.

CODE METRICS

24	25K+	883	100%
CONTRACTS	LINES OF SOLIDITY	TEST CASES	AUDIT COVERAGE

CONTRACT INTERACTION MAP

The following diagram shows how contracts interact with each other:

```
                ┌──────────────────────┐
                │     SBET.sol         │
                │  (Core Entry Point)  │
                └──────┬──────────┬────┘
                       │          │
          ┌────────────┤          ├────────────┐
          │            │          │            │
┌─────────▼────┐ ┌─────▼─────┐ ┌─▼──────────┐  │
│ NFTManager   │ │ Prediction│ │ Integrator │  │
│              │ │ Market    │ │ Hub        │  │
└──────┬───────┘ └─────┬─────┘ └────────────┘  │
       │               │                       │
┌──────▼───────┐ ┌─────▼─────┐           ┌─────▼──────┐
│  NFTVault    │ │ Prediction│           │  Treasury  │
│  (Escrow)    │ │ AMM       │           │  (Hub)     │
└──────────────┘ └───────────┘           └─────┬──────┘
┌──────────────┐                               │
│NFTFeeManager │                    ┌──────────┼──────────┐
└──────────────┘                    │          │          │
                               ┌───▼────┐ ┌────▼───┐ ┌──▼───────┐
                               │Fee Mgr │ │Budget  │ │Vesting   │
                               └────────┘ │Manager │ │Manager   │
                                          └────────┘ └──────────┘

┌──────────────┐                ┌────────┐ ┌────────┐ ┌──────────┐
│ SBETQuery    │                │Yield   │ │Multisig│ │Donation  │
│ (Read-only)  │                │Manager │ │Manager │ │Manager   │
└──────────────┘                └────────┘ └────────┘ └──────────┘
```

DEPLOYMENT ARCHITECTURE

CONTRACT GROUP	DEPLOYMENT ORDER	DEPENDENCIES	UPGRADEABLE
SBET Token	1st (already deployed)	None	No (immutable ERC-20)
Treasury + Modules	2nd	SBET Token	Modules replaceable via hub
SBET Core	3rd	Treasury, FeeManager	No (immutable after audit)
NFT System	4th	SBET Core	Manager upgradeable, Vault immutable
Prediction Markets	5th	SBET Core, Treasury	Market factory pattern
Integrator Hub	6th	SBET Core, FeeManager	Upgradeable via proxy
SBETQuery	Last	All above (read-only)	Freely replaceable

CHAPTER 18

18. Chain Strategy

SBET Protocol is deployed on Ethereum mainnet for token operations and uses Ethereum's security guarantees as the foundation. The protocol's multi-chain strategy focuses on expanding to high-throughput Layer 2 networks while maintaining Ethereum mainnet as the canonical settlement and security layer.

CURRENT DEPLOYMENT

CHAIN	STATUS	CONTRACTS	PURPOSE
Ethereum Mainnet	LIVE	SBET Token (ERC-20)	Token operations, primary liquidity
Ethereum Sepolia	TESTNET	Full protocol (24 contracts)	Development, testing, integration

EXPANSION ROADMAP

PHASE	CHAIN	TYPE	RATIONALE	TIMELINE
PHASE 1	Base	Optimistic L2	Coinbase ecosystem alignment, low fees, growing user base	Q2 2026
PHASE 2	Arbitrum	Optimistic L2	Largest L2 TVL, strong DeFi composability	Q3 2026
PHASE 3	Polygon	Sidechain / zkEVM	Massive user base, gaming ecosystem, low transaction costs	Q4 2026
PHASE 4	Additional L2s	Various	Demand-driven expansion based on user and volume metrics	2027+

CROSS-CHAIN ARCHITECTURE

SBET will use battle-tested cross-chain messaging protocols for token bridging and state synchronization:

LAYERZERO

Omnichain messaging for SBET token bridging (OFT standard). Enables native token transfers between all supported chains without wrapped tokens.

CHAINLINK CCIP

Cross-chain interoperability protocol for secure message passing. Used for cross-chain order settlement and treasury synchronization.

MULTI-CHAIN DESIGN PRINCIPLES

- **Canonical token on Ethereum:** The SBET ERC-20 on mainnet remains the canonical token. All L2 tokens are bridged representations.
- **Independent per-chain state:** Each chain runs its own betting contracts with independent match and pool state.
- **Unified treasury:** Fees from all chains are periodically swept back to the Ethereum mainnet treasury.
- **Chain-specific EIP-712 domains:** Each deployment has its own `chainId` in the EIP-712 domain, preventing cross-chain order replay.

L2 DEPLOYMENT BENEFITS

BENEFIT	ETHEREUM MAINNET	L2 (BASE / ARBITRUM / POLYGON)
Transaction cost	$2-50 per tx	$0.01-0.10 per tx
Confirmation time	12-15 seconds	1-2 seconds
Throughput	~15 TPS	100-4,000+ TPS
Security	Full Ethereum security	Inherits Ethereum security via rollup proofs
User experience	Gas spikes during congestion	Predictable, low-cost transactions

Lower transaction costs on L2s unlock use cases that are uneconomical on mainnet, such as micro-bets (small stake sizes), high-frequency pool adjustments, and real-time prediction market trading during live events.

CROSS-CHAIN SECURITY CONSIDERATIONS

BRIDGE MINIMIZATION

Users can acquire SBET natively on each chain via DEX or CEX, reducing bridge dependency. Only treasury sweeps require cross-chain messaging.

INDEPENDENT STATE

A bridge exploit on one chain cannot affect betting state on other chains. Each deployment is fully isolated with independent match/pool databases.

CANONICAL AUTHORITY

Ethereum mainnet remains the canonical authority for SBET token ownership. L2 tokens are bridged representations backed by mainnet holdings.

REPLAY PROTECTION

EIP-712 domain includes `chainId` and `verifyingContract`. Orders signed on one chain are cryptographically invalid on all other chains.

CHAPTER 19

19. Roadmap

SBET Protocol follows a phased development approach, with each milestone building on the previous one. The roadmap prioritizes security, user experience, and ecosystem growth.

2026 MILESTONES

Q1 2026 — FOUNDATION

Protocol Audit & Core Development

- Versus Security audit completed (146 findings, all addressed)
- Whitepaper v2.0 published
- Frontend dApp with P2P trading, pool betting UI
- AI-powered analytics integration (Claude + GPT-4o)
- Exchange listings: Coinbase, Gate.io, LBank

Q2 2026 — EXPANSION

Base L2 Deployment & NFT Launch

- Base L2 deployment (all 24 contracts)
- NFT betting system live with partner collections
- Integrator Hub launch with first partner onboarding
- Mobile-optimized dApp release
- Prediction markets beta on testnet

Q3 2026 — SCALE

Multi-Chain & Prediction Markets

- Arbitrum deployment
- Prediction markets mainnet launch
- Staking rewards system activated
- Cross-chain token bridging (LayerZero OFT)
- 10+ integrator partners

Q4 2026 — MATURITY

Governance & Ecosystem Growth

- Polygon deployment
- Governance Phase 2 launch (multisig + advisory token vote)
- Treasury yield strategies activated
- API documentation and SDK for builders
- Additional CEX listings campaign

2027 VISION

H1 2027

Full DAO Governance

Transition to Phase 3 DAO governance. Token holders vote on all protocol parameters. On-chain proposal and execution system.

H2 2027

Protocol Ossification

Core contracts made immutable (Phase 4). Second comprehensive security audit. Protocol becomes fully trustless and self-sustaining.

KEY PERFORMANCE INDICATORS

The SBET team tracks progress against the following KPIs at each milestone:

KPI	Q1 2026 TARGET	Q4 2026 TARGET	2027 TARGET
Smart contracts deployed	24 (testnet)	24 (3+ chains)	24+ (5+ chains)
Integrator partners	1-3	10+	50+
Supported sports	10+	20+	30+
CEX listings	3 (Coinbase, Gate, LBank)	5+	10+
Governance phase	Phase 1 (Multisig)	Phase 2 (Multisig + Vote)	Phase 3 (Full DAO)
Audit coverage	100% (Versus Security)	100% + ongoing monitoring	2nd audit + immutability

DEVELOPMENT METHODOLOGY

SECURITY-FIRST DEVELOPMENT

Every smart contract change goes through a three-phase process: design review, implementation with tests, and security analysis (Slither + manual). No code ships without passing all 883+ tests.

PROGRESSIVE DEPLOYMENT

New features deploy to testnet first, then a single mainnet/L2, then expand to all chains. Each stage requires a minimum observation period before proceeding.

COMMUNITY-DRIVEN PRIORITIES

Roadmap priorities are informed by community feedback, usage metrics, and governance proposals. The team publishes monthly development updates.

OPEN SOURCE DEVELOPMENT

All frontend code is open source. Smart contracts are source-verified on Etherscan. Community contributions are reviewed and merged via standard PR process.

CHAPTER 20

20. Risk Factors

Investors and users should carefully consider the following risk factors before participating in the SBET Protocol. The protocol team actively mitigates these risks through engineering, design choices, and operational practices.

RISK MATRIX

RISK CATEGORY	SEVERITY	LIKELIHOOD	DESCRIPTION	MITIGATION
Smart Contract Bugs	HIGH	LOW	Undiscovered vulnerabilities in Solidity code could lead to fund loss	146-finding audit, 883 tests, Slither + Echidna, bug bounty program
Oracle Manipulation	HIGH	LOW	Compromised oracle could report false match outcomes	Multi-oracle validation, finalization timelock, dispute mechanism
Regulatory Action	MEDIUM	MEDIUM	Jurisdictional crackdowns on decentralized betting platforms	Decentralized architecture, no single point of control, progressive DAO
Market / Token Risk	MEDIUM	MEDIUM	SBET token price volatility may impact collateral value	Fixed supply, deflationary burn, multi-CEX liquidity
Bridge Risk	HIGH	LOW	Cross-chain bridge exploits could affect L2 token representations	Battle-tested protocols (LayerZero, CCIP), canonical token on mainnet
Frontend Attacks	MEDIUM	MEDIUM	Phishing, DNS hijacking, or malicious UI could trick users	Open-source frontend, ENS domain, EIP-712 shows data before signing
Liquidity Risk	LOW	MEDIUM	Insufficient counterparties for P2P orders in niche markets	Pool betting for popular events, integrator network, AI market maker
Ethereum Network	LOW	LOW	Network congestion, gas spikes, or consensus failures	L2 expansion reduces mainnet dependency, off-chain order signing

Important: This list is not exhaustive. Cryptocurrency and DeFi protocols carry inherent risks including but not limited to total loss of invested funds. Users should only risk capital they can afford to lose. Past performance and audit completion do not guarantee future security.

RISK MITIGATION FRAMEWORK

SBET employs a layered defense strategy to address each risk category:

TECHNICAL RISK MITIGATION

Smart Contract: 146-finding professional audit, 883 test cases, Slither static analysis, Echidna fuzzing, OpenZeppelin base contracts, CEI pattern enforcement, ReentrancyGuard on all external functions.

ORACLE RISK MITIGATION

Data Integrity: Multi-oracle validation, finalization timelock (community can detect false outcomes), M-of-N dispute mechanism for prediction markets, bonded challenger system.

ECONOMIC RISK MITIGATION

Token Stability: Fixed supply (no dilution), deflationary burn, multi-CEX liquidity (Coinbase, Gate.io, LBank), market making arrangements for healthy order books.

OPERATIONAL RISK MITIGATION

Governance: Multisig for critical operations, daily spending limits, timelock on treasury migration, emergency pause with auto-resume, progressive DAO transition.

EMERGENCY RESPONSE PROTOCOL

SEVERITY	RESPONSE TIME	ACTIONS	AUTHORITY
CRITICAL	Immediate	Emergency pause → investigate → patch → unpause	Any multisig signer (pause); full multisig (unpause)
HIGH	< 4 hours	Assess impact → targeted mitigation → scheduled fix	Multisig majority
MEDIUM	< 24 hours	Investigation → fix in next deployment cycle	Development team
LOW	< 1 week	Tracked in issue tracker → fixed in scheduled release	Development team

INSURANCE & COVERAGE

SBET explores DeFi insurance options to provide additional protection layers for users:

- **Smart contract cover:** Working with DeFi insurance protocols to offer optional coverage against smart contract exploits
- **Treasury reserve:** 15% of treasury operations budget allocated as an emergency reserve for user compensation in extreme scenarios
- **Bug bounty:** Planned bounty program for responsible vulnerability disclosure, incentivizing white-hat security researchers

CHAPTER 21

21. Team & Community

CORE CONTRIBUTORS

SBET Protocol is developed by a distributed team of smart contract engineers, frontend developers, and protocol designers with deep experience in DeFi, sports technology, and cryptographic systems.

SMART CONTRACT ENGINEERING

Solidity development, security architecture, audit coordination. 24 production contracts with 25K+ LoC.

FRONTEND & UX

React/TypeScript dApp, responsive design, wallet integration (RainbowKit + WalletConnect), real-time odds display.

PROTOCOL DESIGN

Tokenomics, fee structure, governance framework, multi-chain strategy, and integrator economics.

AI & DATA

Dual-provider AI integration, confidence scoring algorithms, market analysis, and Kelly criterion optimization.

OPEN SOURCE

SBET's frontend dApp is open source under the BUSL-1.1 license. The smart contracts are source-verified on Etherscan. Community contributions are welcome through the standard pull request process.

COMMUNITY CHANNELS

CHANNEL	PURPOSE
Website	sbettoken.org — Protocol information, documentation, audit report
Twitter / X	Announcements, market updates, community engagement
Telegram	Community discussion, support, trading insights
GitHub	Open-source code, issue tracking, developer contributions

ENGINEERING CULTURE

The SBET development team operates with a set of non-negotiable engineering principles:

SECURITY MINDSET

Assume hostile environments — all inputs, users, networks, and dependencies are untrusted by default. Threat model before implementing. Validate at every boundary.

SIMPLICITY FIRST

Every change should be as simple as possible. Impact minimal code. Find root causes — no temporary fixes. Three lines of clear code is better than a premature abstraction.

INVESTOR WOW

Every meaningful change should improve a visible user journey or metric. Progress should be demoable. Reliability and performance are perceived quality.

DEFENSE IN DEPTH

Authentication + authorization + rate limits + monitoring + isolation + audit trails. No single layer is trusted alone. Fail-closed on unknown states.

TECHNOLOGY STACK

LAYER	TECHNOLOGY	VERSION	PURPOSE
SMART CONTRACTS			
Language	Solidity	0.8.34	Smart contract development with overflow protection
Framework	Foundry (Forge)	Latest	Compilation, testing, deployment, gas profiling
Base Contracts	OpenZeppelin	v5.x	Battle-tested security primitives
Static Analysis	Slither	Latest	Automated vulnerability detection
Fuzzing	Echidna	Latest	Property-based invariant testing
FRONTEND			
Framework	React + TypeScript	18.x / 5.x	Type-safe component architecture
Build Tool	Vite	Latest	Fast development with HMR, optimized builds
Styling	Tailwind CSS v4	4.x	Utility-first CSS with Lightning CSS engine
Wallet	RainbowKit + Wagmi	Latest	Wallet connection, contract interactions, signing
Data Fetching	TanStack Query	5.x	Caching, background refetching (10s stale time)
Validation	Zod	Latest	Runtime schema validation at module boundaries
INFRASTRUCTURE			
RPC	Custom endpoints	—	Configurable via environment variables
Wallet Connect	WalletConnect v2	—	Mobile wallet support via QR code scanning
AI	Anthropic Claude + OpenAI GPT-4o	Latest	Dual-provider analytics, confidence scoring

OPEN SOURCE CONTRIBUTIONS

The SBET team encourages community contributions in several areas:

- **Frontend improvements:** UI/UX enhancements, accessibility, internationalization
- **Integrator tools:** SDKs, example applications, documentation
- **Analytics:** Subgraph indexers, dashboards, data visualizations
- **Testing:** Additional test cases, edge case coverage, fuzzing seeds
- **Security:** Responsible vulnerability disclosure (bug bounty planned)

CHAPTER 22

22. Partnerships & Listings

SBET has established a strong presence across major cryptocurrency exchanges and data platforms, providing liquidity, visibility, and credibility for the protocol.

EXCHANGE LISTINGS

EXCHANGE	TYPE	TRADING PAIRS	STATUS
Coinbase	Tier 1 CEX	SBET/USD	LIVE
Gate.io	Tier 1 CEX	SBET/USDT	LIVE
LBank	Tier 2 CEX	SBET/USDT	LIVE
Bybit	Tier 1 CEX	SBET/USDT	PLANNED
Uniswap	DEX (Ethereum)	SBET/ETH, SBET/USDC	LIVE

DATA & ANALYTICS PARTNERS

PLATFORM	CATEGORY	COVERAGE
CoinGecko	Market Data	Price, volume, market cap tracking
CoinMarketCap	Market Data	Price, volume, market cap, community data
DappRadar	dApp Analytics	Smart contract activity, user metrics, TVL tracking
Etherscan	Block Explorer	Verified source code, token tracker, transaction history

MEDIA COVERAGE

PUBLICATION	COVERAGE TYPE
Forbes	Protocol feature and market opportunity analysis
Crypto news outlets	Listing announcements, protocol updates, milestone coverage

Listing Strategy: SBET's exchange listing strategy focuses on Tier 1 exchanges to maximize liquidity, credibility, and user access. Each listing is paired with market making arrangements to ensure healthy order books and tight spreads.

PARTNERSHIP CATEGORIES

EXCHANGE PARTNERS

CEX and DEX listings provide liquidity access points for users worldwide. Market making arrangements ensure tight spreads and healthy order books on each venue.

DATA PARTNERS

CoinGecko, CoinMarketCap, and DappRadar provide real-time price, volume, and protocol metrics — critical for investor due diligence and discovery.

INFRASTRUCTURE PARTNERS

RPC providers, oracle networks, and cross-chain protocols that form the backbone of SBET's technical infrastructure.

ECOSYSTEM PARTNERS

DeFi protocols, wallet providers, and sports data companies that enhance the SBET user experience and extend the protocol's reach.

LISTING TIMELINE

EXCHANGE	YEAR LISTED	IMPACT
Uniswap (DEX)	2022	Initial decentralized liquidity, accessible to all Ethereum users
Gate.io	2023	First major CEX listing, Asian market access, USDT pair
LBank	2023	Additional CEX liquidity depth, global reach
Coinbase	2024	Tier 1 validation, US market access, institutional credibility

Coinbase Listing: SBET's listing on Coinbase represents a significant milestone, as Coinbase is the largest US-regulated cryptocurrency exchange. This listing provides direct fiat on-ramp access for US users and signals regulatory confidence in the SBET token.

ECOSYSTEM MAP

SBET's ecosystem extends beyond exchanges to include a comprehensive network of infrastructure and data partners:

SBET Ecosystem Partners

EXCHANGE LAYER

CEX: Coinbase (Tier 1, US regulated), Gate.io (global, high volume), LBank (global reach)
DEX: Uniswap (Ethereum mainnet, ETH + USDC pairs)
Planned: Bybit, Base DEXes, Arbitrum DEXes

DATA & ANALYTICS LAYER

Market Data: CoinGecko, CoinMarketCap (price, volume, market cap)
dApp Analytics: DappRadar (smart contract activity, TVL)
Explorer: Etherscan (verified source code, token tracker)

INFRASTRUCTURE LAYER

Wallets: MetaMask, Coinbase Wallet, WalletConnect (50+ wallets via RainbowKit)
RPC: Custom endpoints, fallback providers
AI: Anthropic Claude, OpenAI GPT-4o

MEDIA & COMMUNICATIONS

Press: Forbes coverage, crypto news outlets
Community: Twitter/X, Telegram, GitHub
Documentation: sbettoken.org (whitepaper, audit, API docs)

FUTURE PARTNERSHIP TARGETS

CATEGORY	TARGET PARTNERS	VALUE TO SBET	TIMELINE
Oracle Networks	Chainlink, API3, UMA	Reliable sports data feeds for match finalization	Q2-Q3 2026
Cross-Chain	LayerZero, Chainlink CCIP, Wormhole	Token bridging and cross-chain state sync	Q3 2026
DeFi Insurance	Nexus Mutual, InsurAce	Smart contract coverage for user protection	Q4 2026
Sports Data	API-Football, Sportradar	Real-time scores, statistics, and odds feeds	Q2 2026
L2 Ecosystems	Base, Arbitrum, Polygon foundations	Grants, co-marketing, ecosystem incentives	Q2-Q4 2026
Wallet Providers	Safe, Argent, Rabby	Deep wallet integration, dApp browser features	2027

CHAPTER 23

23. Glossary & Acronyms

KEY TERMS

TERM	DEFINITION
AMM	Automated Market Maker — algorithmic pricing system that provides liquidity without order books
Basis Points	One hundredth of a percentage point (1 bps = 0.01%). SBET prices use 0–10,000 bps (0–100%)
CEI	Checks-Effects-Interactions — Solidity pattern that prevents reentrancy by ordering operations
CEX	Centralized Exchange — traditional exchange with custodial trading (e.g., Coinbase, Gate.io)
CCIP	Cross-Chain Interoperability Protocol — Chainlink's secure cross-chain messaging standard
DAO	Decentralized Autonomous Organization — governance structure controlled by token holders
DEX	Decentralized Exchange — non-custodial trading protocol (e.g., Uniswap)
EIP-712	Ethereum Improvement Proposal for typed structured data signing — enables readable transaction previews
ERC-20	Ethereum token standard for fungible tokens
ERC-721	Ethereum token standard for non-fungible tokens (NFTs)
ERC-1155	Ethereum multi-token standard supporting both fungible and non-fungible tokens
Escrow	Smart contract holding funds/assets until conditions are met
Kelly Criterion	Optimal bet sizing formula that maximizes long-term bankroll growth
L2	Layer 2 — scaling solution built on top of Ethereum (e.g., Base, Arbitrum, Polygon)
LayerZero	Omnichain interoperability protocol for cross-chain messaging and token transfers
LMSR	Logarithmic Market Scoring Rule — prediction market pricing algorithm with bounded loss
Maker	User who creates a P2P order and sets the price
Multisig	Multi-signature wallet requiring M-of-N signers to authorize a transaction
Nonce	Number used once — prevents replay attacks and enables order cancellation
OFT	Omnichain Fungible Token — LayerZero standard for native cross-chain tokens
Oracle	External data source that reports real-world results (match outcomes) to smart contracts
Pari-Mutuel	Betting system where all bets are pooled and payouts determined by pool distribution
Taker	User who fills an existing P2P order at the maker's price

Timelock	Enforced delay between proposing and executing critical actions
TVL	Total Value Locked — aggregate value of assets deposited in a DeFi protocol
Vig / Vigorish	Hidden margin embedded in odds by traditional sportsbooks (typically 5-15%)

CHAPTER 24

24. Legal Disclaimer

GENERAL NOTICE

This whitepaper is provided for informational purposes only and does not constitute financial advice, investment advice, trading advice, or any other form of professional advice. The information contained herein should not be relied upon as a basis for making any investment decision.

NO OFFER OR SOLICITATION

Nothing in this whitepaper constitutes an offer to sell, a solicitation to buy, or a recommendation for any security, token, investment product, or financial instrument. SBET tokens are utility tokens designed for use within the SBET Protocol ecosystem and should not be considered securities, investment contracts, or financial instruments.

RISK ACKNOWLEDGMENT

Cryptocurrency and decentralized finance involve substantial risk of loss. The value of SBET tokens may fluctuate significantly and may go to zero. Smart contracts, despite auditing, may contain undiscovered vulnerabilities. Users should:

- Only use funds they can afford to lose entirely
- Conduct their own research and due diligence
- Consult qualified financial, legal, and tax advisors
- Understand the technical risks of interacting with smart contracts
- Be aware of regulatory requirements in their jurisdiction

FORWARD-LOOKING STATEMENTS

This whitepaper contains forward-looking statements regarding the SBET Protocol's planned development, features, and milestones. These statements reflect current expectations and are subject to significant risks and uncertainties. Actual results may differ materially from those expressed or implied. The SBET team makes no guarantees regarding the completion of any roadmap items or the achievement of any projected metrics.

REGULATORY COMPLIANCE

The regulatory landscape for cryptocurrency and decentralized betting varies by jurisdiction and is subject to change. It is the responsibility of each user to determine whether their use of the SBET Protocol complies with applicable laws and regulations in their jurisdiction. The SBET Protocol is a decentralized technology platform and does not provide or endorse any gambling services.

INTELLECTUAL PROPERTY

The SBET Protocol smart contracts and frontend application are released under the Business Source License 1.1 (BUSL-1.1). The SBET name, logo, and brand assets are the property of the SBET Protocol contributors.

NO WARRANTY

The SBET Protocol is provided "as is" without warranty of any kind, express or implied. The contributors and developers disclaim all warranties including but not limited to merchantability, fitness for a particular purpose, and non-infringement. In no event shall the contributors be liable for any direct, indirect, incidental, special, or consequential damages arising from the use of the protocol.

Important: By interacting with the SBET Protocol, users acknowledge that they have read and understood these disclaimers. Users accept full responsibility for their interactions with the protocol's smart contracts and any resulting financial outcomes.

SBET Protocol

Decentralized Sports Betting on Ethereum

sbettoken.org | Whitepaper v2.0 | March 2026

Contract: 0x2ed2cc2c858a8a8219fd2f2d9e170285dbd02756