Cheshire Terminal Sports

Diagram

```
External Data Sources    Chain Updater        basedchesh API        dApp
   │               │              │           │
   │-- Base RPC         │-- Transaction Logs   │-- Governance     │-- Web3
Interface
                   │-- State Updates

Program Storage:
- Transactions
- Program States
- basedchesh Votes
- Treasury
```

Table of Contents

1. Introduction

Cheshire Terminal Sports Client is a next-generation sports analytics and betting platform built on **Base**. It combines advanced AI prediction models, real-time computer vision analysis, and blockchain technology to create a comprehensive sports intelligence system.

Key Features

- Multi-sport prediction engine
- Real-time video analysis
- Decentralized betting infrastructure
- **basedchesh** token governance
- Advanced AI models

---

2. System Architecture

Data Sources

- Play-by-Play feeds from major sports leagues
- Live tracking data
- Video streams
- Historical statistics
- Player performance metrics

AI Layer

Cheshire Terminal Sports Client AI includes:

- Foundation Model
- Multi-modal data processing
- Pattern recognition
- Historical trend analysis

- Real-time inference

Computer Vision System

- NVIDIA Orin Nano processing
- Player movement tracking
- Game state recognition
- Real-time analysis

Blockchain Infrastructure

- Local **Base** validator
- Exo Labs high-performance cluster
- Smart contract system
- **basedchesh** token integration

---

3. Features

Sports Coverage

- NBA Basketball
- Soccer (Multiple Leagues)
- NFL Football
- Boxing/MMA
- Additional sports being added

Prediction Systems

- Match outcomes
- Player performance
- Real-time odds adjustment
- Risk assessment

- Confidence scoring

Vision Analysis

- Player tracking
- Movement patterns
- Team formations
- Game flow analysis
- Performance metrics

Betting Interface

- Live betting
- Automated market making
- Risk management
- Settlement system
- Performance tracking

4. Getting Started

Installation

```
git clone https://github.com/cheshire-labs/sports-client
cd sports-client
yarn install
```

Configuration

```
{
  "network": "mainnet-beta",
  "cluster": "exo-labs",
  "visionEndpoint": "orin-nano:8080",
  "aiModel": "foundation-v1"
}
```

Running Local Validator

```
base-test-validator
cheshire-sports-client --local
```

5. Components

Foundation Model

The core AI system processes multiple data streams:

- Historical game data
- Real-time feeds
- Vision analysis
- Market dynamics

Vision Processing

NVIDIA Orin Nano handles:

- Live video streams
- Player detection
- Movement analysis
- Event recognition

Data Storage

- Raw data warehouse
- Processed analytics
- Model predictions
- Performance metrics

6. Developer Guide

API Integration

```
import { CheshireClient } from '@cheshire/sports-client'

const client = new CheshireClient({
  endpoint: 'https://api.cheshire.sports',
  wallet: yourWallet
})

// Get predictions
const predictions = await client.getPredictions('NBA')

// Place bet
await client.placeBet({
  match: matchId,
  amount: 1.5,
  odds: 2.0
})
```

Vision Model Integration

```
from cheshire.vision import OrinClient

client = OrinClient()
analysis = client.analyze_stream('game_feed.mp4')
```

7. Governance

basedchesh Token

- Voting rights

- Proposal creation
- Treasury management
- Protocol updates

Voting System

- Proposal submission
- Community voting
- Implementation process
- Treasury allocation

8. API Reference

```
// Endpoints
interface CheshireAPI {
  // Sports Data
  getMatches(sport: string): Promise<Match[]>
  getPredictions(matchId: string): Promise<Prediction>

  // Vision Analysis
  getPlayerTracking(matchId: string): Promise<TrackingData>
  getGameState(matchId: string): Promise<GameState>

  // Betting
  placeBet(bet: BetRequest): Promise<Transaction>
  getMarkets(matchId: string): Promise<Market[]>

  // Governance
  submitProposal(proposal: Proposal): Promise<Transaction>
  vote(proposalId: string, vote: Vote): Promise<Transaction>
}
```

Data Types

```
interface Prediction {
  matchId: string
  homeWinProbability: number
  predictedScore: Score
  confidence: number
}

interface VisionAnalysis {
  players: PlayerPosition[]
  teamFormation: Formation
  possessionStats: PossessionStats
}

interface Market {
  id: string
  odds: number
  liquidity: number
  volume: number
}
```

For more detailed documentation and updates, visit our GitHub repository or join our Discord community.

Additional Notes

Sports Data Sources

- NBA Stats API
- Soccer APIs (for multiple leagues)
- NFL/Football Data
- Combat Sports Data (Boxing/MMA)
- Each source provides real-time and historical data

AI Prediction Engine (New Component)

- **Historical Analysis**: Processes past performance data
- **Real-time Updates**: Incorporates live game/match data
- **ML Models**: Different models for each sport type
- **Odds Generation**: Creates betting lines and predictions

Chain Updater

- Now handles multiple sport types
- Processes predictions and betting outcomes
- Updates on-chain states for all sports markets

Program Storage

- Sports Data: Match results, player stats, league data
- Predictions: AI model outputs and performance tracking
- Transactions: Betting history and payouts
- Governance: **basedchesh** token voting and treasury

dApp Interface

- **Sports Dashboard**: Views for all supported sports
- **Betting Interface**: Place bets across all sports
- **Governance Portal**: **basedchesh** token holder functions

Central AI Agent

- Processes data from multiple sports simultaneously
- Generates sport-specific predictions
- Learns from historical performance
- Adjusts odds in real-time based on events and betting patterns
- Provides confidence scores for predictions

Cheshire Terminal Sports Client — Where advanced AI, real-time vision analysis, and decentralized betting come together on **Base** with **basedchesh** token governance.