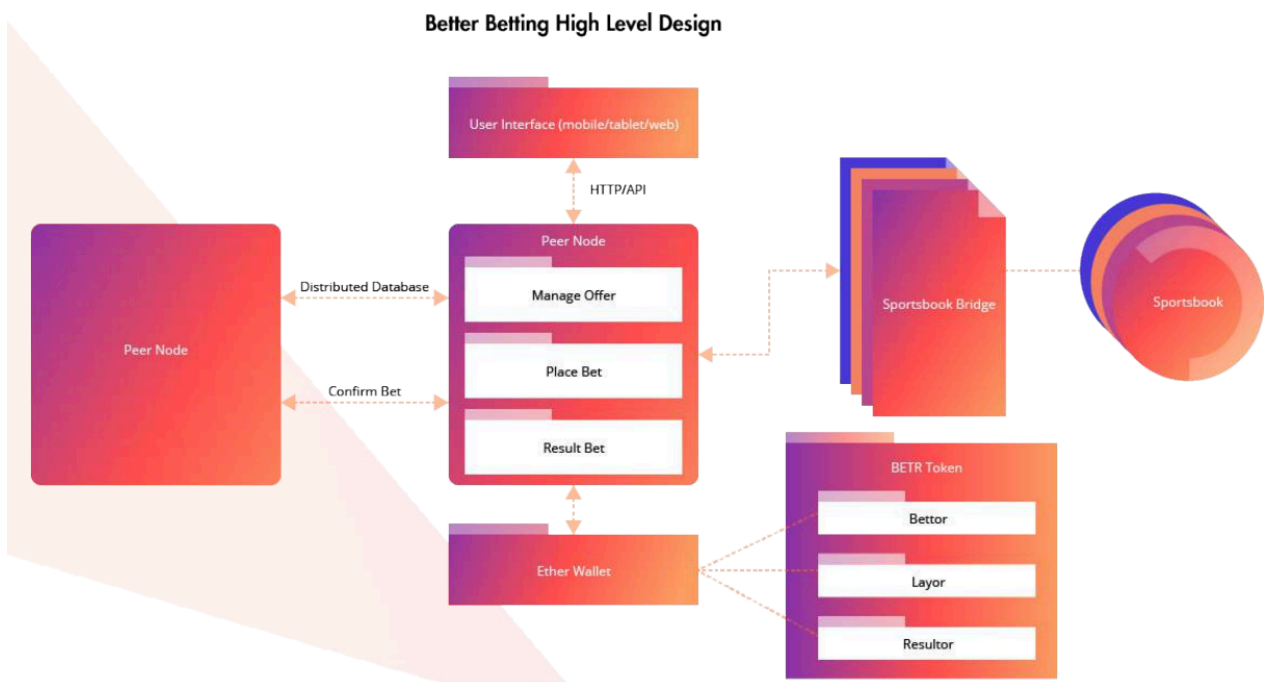




## Better Betting API version 3.08

This document outlines the API for placing a bet via the Better Betting system.



There are a number of components in the overall system. This document contains a first draft of the API that will be offered by any instance of the Better Betting Node (BBN) to facilitate an interface with either the sportsbook bridge or a UX to facilitate betting. The BBN is referred to in the above diagram as the Peer Node.

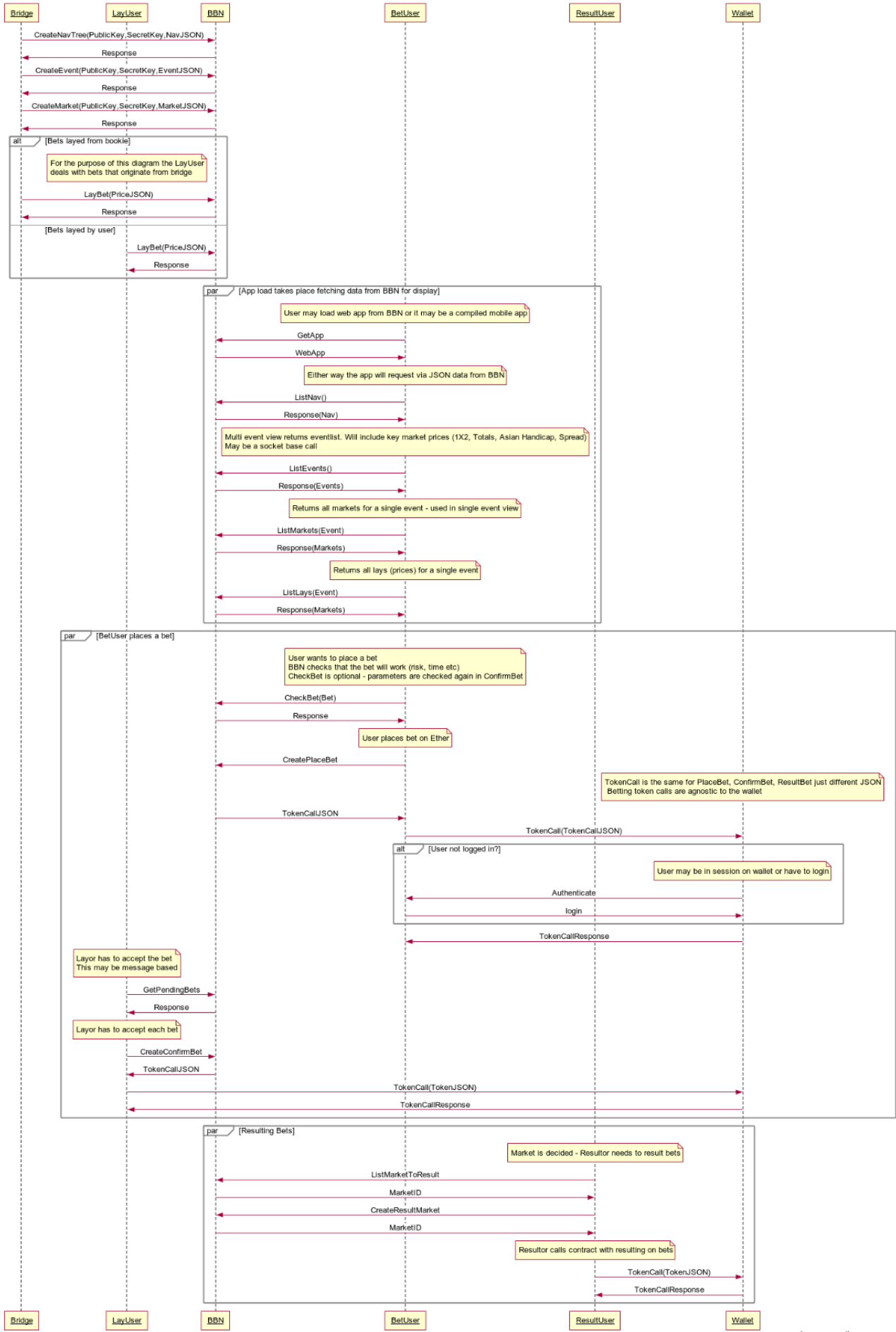
The sportsbook bridge is a distinct component that provides a link between existing data providers and/or sportsbooks and maps the data including lay-data from one to the other. It may incorporate its own Ethereum wallet in order to automate the ConfirmBet and ResultMarket API calls which take place directly on Ethereum. Note that these calls can also be invoked manually through interaction with any compliant Ethereum wallet to place authenticated token method invocations.

Markets and events will be supplied by Better Betting for the World Cup. Future iterations of this API will be enhanced with the ability for the API to create markets and events

### Method of placing a bet

The following sequence diagram outlines the methodology for placing a bet.

### BBN Interaction



## Public Methods

---

### ListNav

returns id-s of root NavElement-s

---

### ListEvents

The first pass will only support request by NavElement : World Cup. Other filtering will be done by the client side javascript.

#### Interface

filters:

- by countryID
- by participantID
- by navElementID
- by marketID
- string timeRangeID

returns a list of all Events with participants, event time, parent ID, markets.

bad filter option

---

### ListMarkets

- by NavElementID
- by eventID

Returns markets supported

---

### ListLays

required uint layerID

filters

- by NavEntryID
- by marketID
- by participantID
- by resolverID
- by layerID
- by eventID

returns a list of layed Bets in a JSON format

---

### ListKnownBBNs

returns a list of BBN addresses

---

## Authenticated Methods

In general authenticated methods have to be invoked through a token call on Ethereum although not all result in a transaction on Ethereum.

---

### LayBet

Lays a bet onto the BBN network on the behalf of the user. This is an authenticated method – authenticated via Ethereum authentication.

#### Interface

- required uint EventID
- required object[] LayData
- required string ResultorID
- required string LayerID
- optional object Params

#### LayData = array of

- required uint marketTypeid
- required object[] selections = array of
  - required uint selectionId
  - required string risk
  - required string odds
  - required string status (offered/removed/suspended)

LayerID is Ethereum public wallet address of the layer

#### returns

- BBN not active
- BBN returns OfferID (generated by the BBN)
- BBN too busy
- rejected unknown data

---

### CreatePlaceBet

Creates JSON token call to reserve a bet on behalf of the Bettor. This would be followed by ConfirmBet.

#### Interface

- required string OfferID
- required uint EventID
- required object LayData
- required object PlaceData
- required string LayerID
- required string BettorID
- required string ResultorID
- optional object Params

#### LayData = array of

- required uint marketTypeid
- required object[] selections = array of
  - required uint selectionId

required string stake  
required string odds

LayerID is Ethereum public wallet address of the layer

PlaceData = array of  
required uint marketTypeid  
required object[] selections = array of  
    required uint selectionId  
    required string stake  
    required string odds

BettorID is Ethereum public wallet address of the bettor

returns  
JSON method call for token  
Rejected unkown offerId

---

## CreateConfirmBet

Creates JSON token call to confirm the bet placed by the bettor with PlaceBet call.

interface  
    required string OfferID  
    required uint EventID  
    object LayData  
    object PlaceData  
    required string ResultorID  
    required string BettorID  
    required string LayerID  
    optional object Params

LayData = array of  
    required uint marketTypeid  
    required object[] selections = array of  
        required uint selectionId  
        required string stake  
        required string odds

LayerID is Ethereum public wallet address of the layer

PlaceData = array of  
    required uint marketTypeid  
    required object[] selections = array of  
        required uint selectionId  
        required string stake  
        required string odds

BettorID is Ethereum public wallet address of the bettor

returns  
JSON method call for token  
rejected unknown data

---

## ListMarketToResult

Returns a list of markets to result  
Interface

required resultorID  
optional marketID

Returns Array of  
int marketID  
string smartContractIndex

---

## CreateResultMarket

Creates JSON token call to result a market on behalf of the resultor.

Interface

required int marketID  
required int resultorID  
required string smartContractIndex  
required string bettorpayoutamt  
required string layerpayoutamt  
required string resultingOptions (not req?)  
required string cryptSignature

returns

accept  
smartContractIndex not found  
marketId not found

---

## Type Definitions

### UINT EVENTID

Internal Better Betting ID for a specific event. These are fixed values and do not change. For the purposes of external IDs they are matched to internal IDs via a translation layer.

Only non-negative values.

### UINT MARKETID

Internal Better Betting ID for a specific market. These values are correlated directly with a smart contract in the block chain but are not mapped directly.

These are fixed values. For the purposes of external IDs they are matched to internal IDs via a translation layer. Only non-negative values.

**STRING LAYERID**

Wallet ID of the layer

**STRING BETTORID**

Wallet ID of the Bettor

**STRING NAVELEMENTID**

Top level navigation element

**STRING SMARTCONTRACTINDEX**

Index into the smart contract array for the instance of the specific bet.

**STRING STAKE**

Amount of risk the party is willing to accept. This is a string value, non-negative.

**STRING PRICE**

The price associated with the layed bet. Stringified decimal values.

**STRING OPTIONOFFERED**

Option offered by the layer on a specific market. For example in 1x2 it would be one of 1:participantID,x, or 2:participantID.

**STRING RESULTORID**

Text value of the chosen resultor set by the layer. This can be the original layerID but not necessarily always equal. A 3rd party resultor may be nominated.

**STRING LAYERID**

Text value of the Layer party of a Bet.

**STRING TIMERANGEID**

Stringified value of a time range, or specific time.