Serpent contracts on Counterparty, we are also testing an alternative implementation of Augur using Serpent. Counterparty's announcement frees us to use Bitcoin as the transactional currency on our platform, instead of having to use Ethereum's ether – this will also be remedied on Ethereum itself when sidechains are released.

We think smart contracts may provide an alternative method for implementing Augur. However, there are several potential barriers related to security and scalability. We are currently testing our contract implementation and will update the whitepaper with our results.

Appendix C: LMSR Derivation

Here we present a simple derivation of the logarithmic market scoring rule (LMSR), using the principle of Maximum Entropy [22–24].

Let q_i denote the number of shares outstanding of outcome i, and $P(q_i)$ denote the *probability* that there are q_i outstanding shares of outcome i. The *entropy* of the market is given by the standard form [14, 25]:

$$S(\{q_i\}) = -\sum_i P(q_i) \log P(q_i), \tag{C1}$$

where the sum is over all possible outcomes in this market. Suppose there are two constraints on the entropy. The first is the mean number of shares outstanding, across all outcomes in the market,

$$\langle q \rangle = \sum_i q_i P(q_i), \tag{C2}$$

and the second is normalization:

$$\sum_i P(q_i) = 1. \tag{C3}$$

The market's Lagrangian is $\Lambda(\{q_i\}) = S - \langle q \rangle - 1$. The constrained maximum entropy is found where the derivative of Λ vanishes,

$$\mathrm{d}\Lambda = \sum_i \mathrm{d}P(q_i) \Big[\log P(q_i) + 1 + \alpha + \beta q_i \Big] = 0, \tag{C4}$$

where the multipliers α and β enforce normalization and constraint C2, respectively. Since Eq. C4 is required to hold for all possible i, it simplifies to:

$$\log P(q_i) + 1 + \alpha + \beta q_i = 0. \tag{C5}$$

Using Eq. C3 to eliminate the normalization parameter $\mathrm{e}^{-1-\alpha}$, reveals the Boltzmann distribution:

$$P(q_i) = \frac{\mathrm{e}^{-\beta q_i}}{\sum_j \mathrm{e}^{-\beta q_j}} = Z^{-1}\, \mathrm{e}^{-\beta q_i}, \tag{C6}$$

where Z is the canonical partition function:

$$Z = \sum_i \mathrm{e}^{-\beta q_i}. \tag{C7}$$

As in statistical physics, Eq. C6 describes a family of probability distributions, characterized by the value of the Lagrange multiplier β. The value of β, which is set by *data*, is the connection between the information-theoretic derivation and reality. In thermodynamics, β is the inverse of the system's temperature; for a market, it is set by the market's liquidity. Examining Eq. C6 and Eq. 4, it is clear that β is set by the LMSR's loss limit parameter (ℓ),

$$\beta = -\frac{1}{\ell}, \tag{C8}$$

we see that Eq. C6 is identical to the LMSR price function (Eq. 4):

$$p(q_i) = P(q_i) = Z^{-1}\, \mathrm{e}^{q_i/\ell}. \tag{C9}$$

Eq. C9 is the result expected for a prediction market: the *price* offered for shares of outcome i is exactly equal to the *probability* that outcome i will occur.

In this framing, the number of shares q_i is analogous to a thermodynamic *energy* (or internal energy). The LMSR's cost function is just the logarithm of the sum of statistical weights, which we have identified as the system's partition function (Eq. C7).

$$C(\{q_i\}) = -\ell \log Z = \langle q \rangle + \ell S. \tag{C10}$$

This is identical to the Helmholtz free energy, a thermodynamic function which describes the amount of work required to generate the system's observed statistical configuration. This configuration is the distribution of energy levels for a thermodynamic system; for a market governed by the LMSR, it is the share distribution.