

# Rakugo Technical Whitepaper

Wyatt Meldman-Floch

July 14 2017

## Abstract

Rakugo revolutionizes online publishing through notarization and monetization on the Ethereum blockchain. When a user publishes content through Rakugo, they are paid royalties for revenue generated by their content and the content is copyrighted through notarization on the Ethereum blockchain. The Rakugo Engine goes further by optimizing royalties through increased user influence and trend creation.

## 1 Rakugo Platform

The functionality of the Rakugo platform abstracts crowdsale analytics and interaction with the Ethereum blockchain with an intuitive user interface. Our approach is a web application that combines content management with the functionality of an ethereum wallet. Revenue is generated from published content and distributed to holders of tokens representing online content. Content producers can choose to have funds transferred as USD or as ETH via our revenue distribution model outlined below.

Rakugo creates a digital representation of digital content on the Ethereum blockchain by deploying an ERC20-standard token-creating smart contract with a unique digital fingerprint. Content is notarized by the deployment of the contract and ownership of the content is represented by the ownership of these tokens. The parameters of token creation and ownership structure are supplied by the creator during deployment of the smart contract. Tokens can be traded or held via an ownership model of the creator's choosing. Revenue is generated when content is published through Rakugo and is dispersed based upon the owner's chosen configuration.

The ways in which a creator can issue representations of their content are boundless and Rakugo empowers users to try anything without the burden of writing or deploying code to the Ethereum network. Rakugo empowers all

creatives with the ability to notarize and monetize their work with the security and accessibility of a distributed ledger that blockchains provide.

## 2 Rakugo Assetization

The crux of Rakugo's value is our ability to "assetize" or create a digital representation of content on the Ethereum blockchain. These representations are referred to as 'Namespace Tokens', as their publicly accessible name attribute is composed of a logical grouping of unique identifiers.

ERC20-standard 'Namespace Tokens' are created via smart contract, each with an identifying payload that notarizes the content and its creator. The following JSON schema describes information that will be available after de-compression of a token's digital fingerprint attribute

```
{
  "$schema": "digital_fingerprint",
  "definitions": {},
  "id": "https://rakugo.co/whitepaper",
  "properties": {
    "compressedContent": {
      "id": "/properties/compressedContent",
      "items": {
        "id": "/properties/compressedContent/items",
        "type": "integer"
      },
      "type": "array"
    },
    "link": {
      "id": "/properties/link",
      "type": "string"
    },
    "name": {
      "id": "/properties/name",
      "type": "string"
    },
    "publishDate": {
      "id": "/properties/publishDate",
      "type": "string"
    }
  },
  "type": "object"
}
```

A JSON object corresponding to the schema above is compressed and added as a public attribute of the token creation contract as the pseudo-solidity code below

```
contract MyToken {  
  
    string public name;  
    bytes public digitalFingerprint;  
  
    function MyToken(...) {  
        name = "wyatt.meldman-floch.rakugo.whitepaper";  
        digitalFingerprint = /compressed json bytes/;  
  
        ...  
    }  
}
```

### 3 Rakugo Revenue Distribution

The Namespace Tokens described above are the tool used to redeem revenue from the Rakugo platform. Revenue (at the owner's choice) can be disbursed to token holders via a smart contract where the token(s) is used to attribute the correct allocation of revenue to its holder. This decentralized approach allows users to freely manage their assets without relying on the Rakugo platform if they so choose. Users also have the ability to automate the revenue distribution through our the Rakugo platform.

This model provides a safer approach to revenue distribution than similar Dapps that rely on a third-party API (such as a block explorer for finding wallet addresses of token holders). Distribution occurs via a handshake where a token(s) is sent to a smart contract and revenue along with the token(s) is sent back to the token holder. Thus if this wallet address is not hosted through our platform we can verify ownership and safely disburse revenue to the rightful owner.

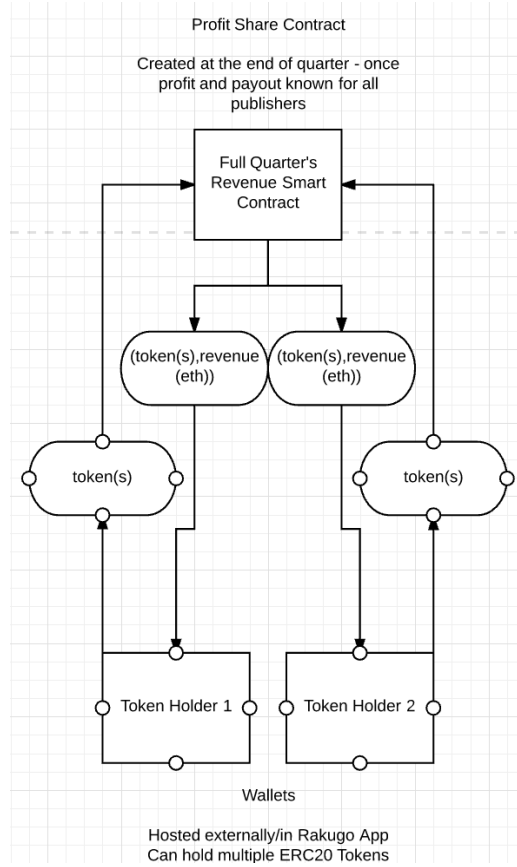


Figure 1: Revenue Distribution Architecture

## 4 Smart Publishing

Rakugo employs a "Smart Publishing" approach to revenue optimization which prioritizes relevance to trending topics on top search engines. Tailoring content to match trending topics increases search engine exposure, revenue generation and overall influence of the content producer. Content is matched to currently trending topics via a proprietary topic modeling algorithm and suggestions that increase relevance to these topics are provided to the producer. Beyond that, Rakugo seizes opportunities to drive emerging trends, increasing the overall influence and value of the Rakugo platform.

Trends are identifiable across streams of data using a cross dependence tem-

poral topic model (CDTTM)<sup>1</sup>. A cross dependence temporal topic model identifies dependent and temporal correlation across streams of data for building correlated generative processes with mutual reinforcement.<sup>2</sup>

Consider topic distributions,  $\theta_1, \theta_2$  across streams of major news outlet content and Rakugo user generated content (UGC) where  $\phi$  denotes the word distribution of each topic;  $x$  is a binary variable indicating whether the generation of the current word is influenced by the previous news or UGC ( $x = 1$ ) if so or ( $x = 0$ ) if not.  $\alpha, \beta$  are the Dirichlet hyper parameters and  $\lambda$  is the Bernoulli parameter for sampling  $x$ ,  $\gamma_d, \gamma_p$  are its hyper parameters.

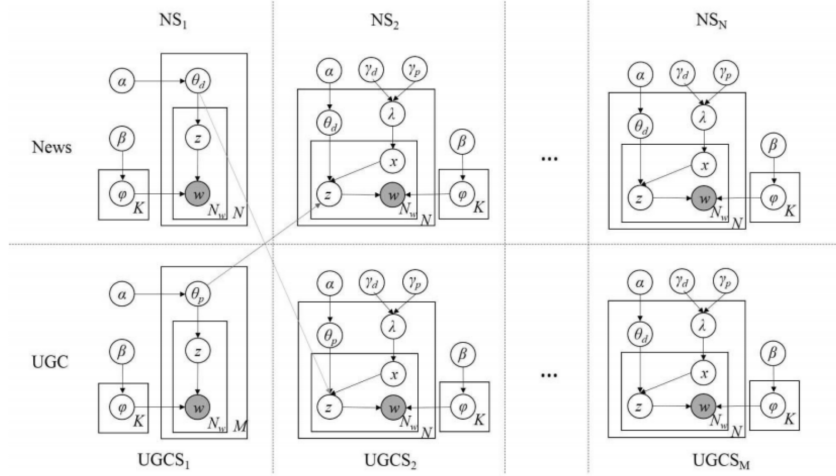


Figure 2: Cross Dependant Temporal Topic Model<sup>1</sup>

For the generative process, we partition both streams into disjoint substreams with fixed time intervals, e.g.  $NS = NS_1 \cup \dots \cup NS_n$  where  $NS_i$  is with time  $[t_i, t_i+1)$ , and initialize two standard LDA models in the first substreams  $NS_1$  and  $UGSC_1$ . As for the subsequent substreams, if there is a previous substream in the other stream (e.g.  $NS_{i_1}$  has a previous UGC substream  $UGSC_{i_1}$ ), a coin  $x$  is tossed according to  $p(x|d) \sim (\gamma_d, \gamma_p)$  to decide whether  $w_d$  inherits from the previous substream, otherwise (namely when there is no previous substream in the other stream) a standard LDA model can be employed for word sampling. The diagram in Figure 2 graphically describes this algorithm. Further

<sup>1</sup>Wang, X., Zhai, C., Hu, X., Sproat, R.: Mining correlated bursty topic patterns from coordinated text streams. In: Proceedings of the 13th ACM International Conference on Knowledge Discovery in Data Mining, pp. 784–793 (2007)

<sup>2</sup>Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 113–120 (2006)

descriptions of similar CDTTM implementations can be found in (Wang, X. et al).

A CDTTM allows us to quantify topical influence across streams using a common metric in information theory known as Kullback-Leibler(KL) divergence.<sup>3</sup> If topics  $z_1$  and  $z_2$  associated with two distributions  $\psi_1$  and  $\psi_2$ , influence  $\zeta_{(z_1 \rightarrow z_2)}$  from  $z_1$  to  $z_2$  can be defined as the added information content in  $z_1$  compared to  $z_2$ .

$$\zeta_{(z_1 \rightarrow z_2)} = KL(z_1|z_2) = \sum_{i=1}^{\nu} \psi_{2i} \times \log \frac{\psi_{2i}}{\psi_{1i}}$$

Rakugo increases individual influence in two ways: by tailoring individual content to trending topics via a proprietary topic-document similarity score and by optimizing Topic Progressiveness, a metric denoting the possibility of user content triggering trends in major news channels, between the stream of all Rakugo articles with respect to top content sources. For each topic  $z_p$  in the Rakugo content stream with a corresponding linked topic  $L(z_p)$  in a major content stream, the cross-entropy values are computed to find the minimum  $z_d^{min}$ , and then the Topic Progressiveness is defined as:

$$Prog(z_p) = T(\underset{z_d \in L(z_p)}{\operatorname{argmin}} H(z_p, z_d)) - T(z_p)$$

where  $H(z_p, z_d)$  is the cross-entropy of  $z_p$  to its linked topic  $z_d$  and T denotes the timestamp of the input topic.<sup>3</sup> Optimization of Topic Progressiveness increases influence of the individual user and the total influence of all Rakugo generated content.

## 5 Rakugo Smart Publishing Engine

The Rakugo Smart Publishing Engine recommends edits to content based upon trending topics identified in real-time. Using the methods described above, our engine performs real-time topic modeling between Rakugo and competing content streams. From the user's perspective we offer on demand edit suggestions to align sentiment to trending topics. A topic similarity score is generated by our matching algorithm between content and trending topics, suggestions to increase topic relevance are then provided to the user. From a macro perspective,

---

<sup>3</sup>Hou, L., Li, J., Li, X.L., Su, Y.: Measuring the influence from user-generated content to news via cross-dependence topic modeling. International Conference on Database Systems for Advanced Applications, 125-141

our engine provides suggestions to all Rakugo users when the potential to influence the emergence of a trending topic presents itself. Using historical data and currently trending topics identified by our CDTTM platform, edit suggestions are provided in the same manner as our topic relevance score and users are rewarded for their contribution to resultant trends identified by the CDTTM platform after the trend creation campaign.

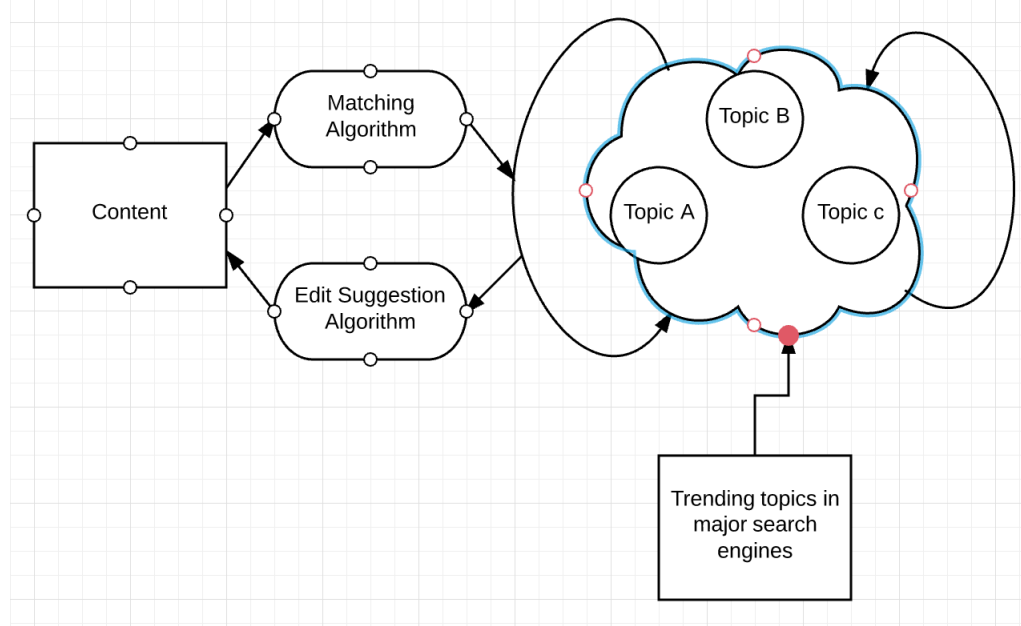


Figure 3: Content Editing Architecture

## 6 Conclusion

The Rakugo Platform combined with our smart publishing engine provides a next generation approach to content management and publishing. Unlike other applications like Medium, Buzzfeed or Steemit content creators own the rights to their content through copyright and get paid their fair royalties as actual money. Rakugo goes beyond that by optimizing royalty generation, influence and the ability to monetize or share value as the user chooses. These details will be abstracted by an intuitive user interface but will also be accessible to power users via public a API.