# Bit Elegy

Brian Wing Kui Ng

Whitepaper: May 10, 2018

## 1 Prospectus

We have killed poetry, and we can prove it.

1. Poetry is the function of utterance that exists for its own sake.[1]
2. A new poem entitled *Bit Elegy* by Brian Ng is distributed over a decentralized Solidity smart contract on the Ethereum network.
3. The poem is publicly disclosed only when a valid payment is sent. Otherwise, the poem will have been inaccessible to anyone but the poet.
4. - If the poem is paid for, the poem will have been uttered for its utilitarian function.
   - If the poem is not paid for, the poem will never have been uttered.
5. The remains of this document will be non-poetic.

You, regardless of whether you make a bid or not, are responsible for this ritualized murder of art. More will follow.

## 2 Poetics: A Quantity Theory of Non-Poetry

In the famous story, "Pierre Menard, Author of the Quixote," Borges describes a character who has authored a text line-for-line identical to Cervantes' "Don Quixote." This text, the protagonist stresses, is not a copy. By virtue of it being written in the present day; it differs from Cervantes' in that it now bears the history of the interloping years, bears the subtler imagination of its writer. This was the spirit in which Bit Elegy was produced: can the poetics of a document be destroyed even with, provably, no change in its semiotic features?

We make here a difference between killing as putting to end, and as annihilation. The difference between these two theses is that annihilation is not a decisive schismatic. In our ritual the choice to pay or not to pay does not confer guilt to any party because all bear the responsibility of the killing. Neither do we claim that we are the first or only to kill poetry. Nor do we claim that all poetry that has been published for a profit is dead. Nor do we implicitly claim that the essence of poetry's death is technological or financial. The contribution of this ritual is only to (viz. Laruelle's concept of non-photography) prove a non-specular, decisionless move from making to non-making, from poem to utterance.

[1] cf. the dictum of *Art for art's sake*; Jakobson's poetic function in *Linguistics and Poetics* (1960).

Now that we have designed a ritual around poetry's remains, what do we do with this non-poetic document? For one, we question the burden to read texts as if they could only exist for themselves. In "The Uprising: On Poetry and Finance" Franco Berardi makes an analogy between the social contracts of the state apparatus and the poetry as a model of semiotic transactions, and the deterritorialization of both by finance capitalism. (One could imagine Bit Elegy as an anti-security: without declared ownership, untransferable without being destroyed.) By creating and burning alternative assets, by infiltrating the superstructure and aiming at the tautological, self-serving nature of finance capitalism, we can perhaps imagine something outside the catachronistic conditions in which we all are precarious.

## 3 Praxis

"Bit Elegy," a poem dedicated to deceased poet John Ashbery (1927 - 2017), is a document of 10 pages on Letter format (8.5" by 11"), consisting of 2682 words and 15408 characters, revised from 10/24/2017 to 3/27/2018, typeset in 12pt Nunito font on a PDF.

The bulk of this text was written from 11/8/2017 to 11/15/2017, and roughly half of that on a caffeinated writing session with Alex Karsavin on 11/8/2017 across two cafes in Bridgeport, Chicago.

The poet calculates the SHA-256 hash of the PDF for Bit Elegy with the bash command:

```
openssl sha -sha256 BIT_ELEGY.pdf
```

which returns a hash value of

```
cd3fd477471a7d0947074648bbe1efedbfffebbf177981fac0d26fb570abb97
```

Given that the probability of another file having the same hash is 4.3e-60, and few of all possible files are PDFs that can be successfully opened, this hash is identifiable as uniquely associated with the poem without revealing its contents. On 3/26, the poet appended this hash to a Solidity contract onto the Ethereum blockchain, deployed at address

```
0xaab9a0d24ba52fbc86c9a93e3915d656dcac0f0f
```

whose contents are follows:

```
    pragma solidity ^0.4.21;

/** @title Bit Elegy: Solidity contract for a piece by Brian Ng. */
contract BitElegy {
    address public poet = msg.sender;
    bytes public textHash =
    "18ebb4e0f81bc7200a6e2e95ccff5941fccbbb56e2e854f91106b4820888c7e5";

    address public killer;
    bool public isKilled = false;


    /** @dev This function kills the poetic function of the text,
     * "Bit Elegy," given that a killer, who is not the owner of the contract,
     * sends more than 10 finney to the owner.
     */
    function killPoetry() external payable {
        require(!isKilled);
        require(msg.sender != poet);
        require(msg.value > 10 finney);

        // "Tear him for his bad verses! Tear him for his bad verses!"
        poet.transfer(msg.value);
        killer = msg.sender;
        isKilled = true;
    }

    /** @dev Fallback function if someone really wants to tip me. */
    function () external payable {
        poet.transfer(msg.value);
    }

}
```

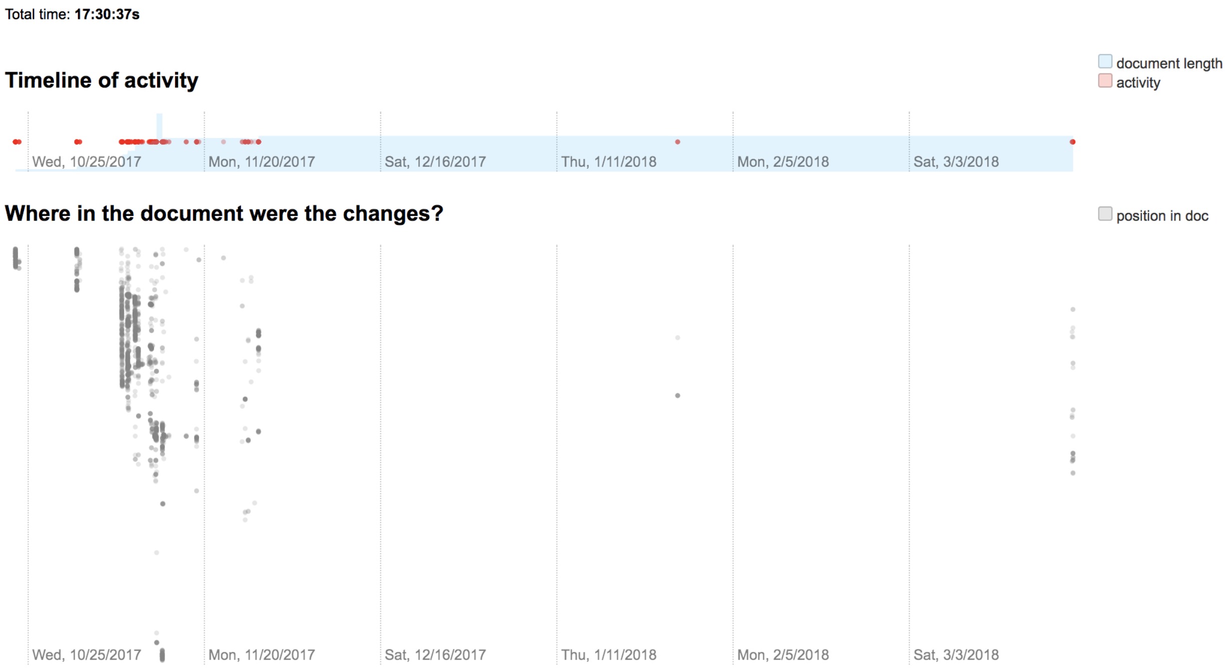
In plainer terms, what this contract does is:

- Uniquely identifies that the poet has a poem of some hash;
- Given that the poem is not killed, that the message sender is not the poet, and that the message contains a value of more than 0.01 ETH, declare the poem killed.

Within 22 hours, three payments were received totalling a little more than 0.04 ETH (roughly $30 as of writing). We declare the agent first to send money, associated with wallet `0x792ba9332f2dfccbba3055d98bbb9a66f0ce31e0`, the killer.

## 4 Analytics

Using Draftback, we present a genetic history of editorial progress on Bit Elegy:



## 5 Security Breaches

Significant portions of this document were, in possible violation of the stringent requirements of the experiment, accessibly uttered to members of the public preceding the killing of the poem. I list them here in the spirit of the Dogme 95 film collective's confessions:

- The bulk of this document was composed on Google Docs. The content of G Suite services, Google claims, "is encrypted using 128-bit or stronger Advanced Encryption Standard (AES)." However, the poem may have been intercepted if the poet's authentication information was unknowingly leaked in any form during the writing of this poem. Moreover, Google's policy for legal subpoenas in the United States is to "notify the user via email before any information is disclosed unless such notification is prohibited by law." While the poet received no such email, nor any notification of a security breach on his G Suite Account between the date the poem was begun and its killing, it is possible that said document could have been illicitly viewed by an attacker or a state agency.

- Parts of the poem were verbally shared with Alex Karsavin during a writing session in Bridgeport Coffee Company on 11/8/2017.

- Two stanzas, beginning "'I gain,' the understudy writes to his mother, 'more power by the day.'" to "Twitches on his leg, as if embarrassed of his helpless knowledge of its inquiry" are lifted from an unpublished but previously circulated poem, "Zero Mostel Entertains the Troops."

- Four quatrains, beginning "When the mouth dies, who misses you?" and ending with, "Cech cohomology commutes for quasicoherent sheaves" were shared with the critic Daniel Tiffany via Facebook message on 7/24/2017.

- An agreement over a Facebook post to swap a sentence with the description of ”a loose adaptation of Freud’s =Dora= as a royal drama starring a young Charlotte Gainsbourg as Ida Bauer and Julianne Moore as a governess/therapist” on 3/13/2017 with critic Robert Archambeau resulted in the line, ”I’ve bought season tickets to your dreams.”

While several lines are interpolated quotations (the aforementioned ”When the mouth dies, who misses you?,” for instance, is a quotation from Berryman’s ”Homage to Mistress Bradstreet”; various Ashbery poems are referenced) the poet does not view them as security breaches, as these borrowings are altered by the poem’s juxtapositions and context.

In addition, the smart contract has a bug of sorts: the first transaction sent money without explicitly calling function killPoetry(). Hence the address of the killer on the smart contract does not reflect the first agent who sent money.

## 6 Acknowledgements

I am thankful for productive conversations with (among others) Alex Karsavin, Alston Boyd, Mara Iskander, Dani Doerr, Brian Kim Stefans, Jeremy Cahill and Travis Dean Uhrig. I am encouraged by the rapidly expanding tooling constructions for contract development (e.g. Remix, Truffle, web3) which made the technical aspects easier to develop in early 2018 than late 2017.