It seems to me that BLS signature aggregation can be easily used as a very efficient accumulator of signed messages. If this is already widely known, I apologize - but I haven't run into this before. Here is how it could work:

Suppose:

- We have a set of n

messages m_1....m_n

(these could be votes from n

validators).

- Each message is signed using BLS signature scheme such that S_i = p_i \cdot H(m_i)

, where p_i

is a private key, and P_i

would be a corresponding public key.

We can then use a simple BLS aggregated signature as an accumulator of tuples (m_i, S_i)

. Specifically:

A = S_1 + S_2 + ... + S_n

For such an accumulator, a witness needed to prove inclusion/exclusion is just the public key P_i

. So, given a witness, we can check if a given tuple (m_i, S_i)

is in the accumulator by checking:

e(G, A - S_i) = \frac{e(G, A)}{e(P_i, H(m_i))}

This accumulator also has some nice properties:

- Since it's just a BLS signature, it's less than 100 bytes in size.
- Witnesses are just the public keys - so, also less than 100 bytes in size.
- Adding/removing values to/from the accumulator is just adding/subtracting signatures - so, very easy.
- Moreover, the witness remains constant even as values get added or removed to/from the accumulator.
- Two non-overlapping accumulators can be aggregated just by summing them.

The obvious drawback is that this accumulator can hold only signed messages (not arbitrary messages), but I think there are plenty of examples in crypto when we deal with signed messages.