The purpose of this topic is to collectively agree on the scope for v2 of the resource machine report, most likely coauthored by @vveiln and myself (a wee bit).

Most ideas here are sourced from this thread - I suggest that we pick the most relevant ones under a few criteria:

1. Converging the interfaces to what we expect to be the final shape as fast as possible, so that we can aim to provide application API stability later this year.
2. Clarifying calling conventions and all points relevant to developer UX.
3. Working out & clarifying real examples to aid in both (1) and (2).

Current proposed scope (as an extension to what's already in v1):

- Balancing signature folding proofs (see here)
- Distributed partial evaluation analogy (see here), times of execution, and details on how transaction candidates may be combined pre-ordering.
- Support for non-linear resources, and example usage as oracle data (e.g. timestamp).
- Information flow control for transactions / transaction candidates, and related details on upgrading / downgrading proofs, information revelation, etc.
- Clarify all calling conventions.
- Additional examples of:
- Non-linear resources
- Information flow control & visibility to different parties
- Account abstraction
- Non-linear resources
- Information flow control & visibility to different parties
- Account abstraction

Folks who should definitely read over this / give input:

- @mariari & @ray from the developer UX perspective
- @isheff from the Typhon perspective - when / how do we want to integrate controllers here?
- @paulcadman and Juvix folks from the Juvix perspective - are there parts of the API which you would like to be clarified or changed?