The Kinds Of Problems Solved By SUAVE

These four categories cover all SUAVE use cases that we have come up with so far (I think). It's worth noting they are all about increasing the trust between counterparties based on one side being able to make stronger commitments by running the TEE. These are all examples of SUAVE making it easier for one party to trust another. Its interesting to think why the examples we listed here are not served by blockchains today. The answer in many cases is privacy, current smart contracts give certain guarantees, but can't really control how information is used (especially not in an efficient generic way).

caution: this isn't BD work or anything like that - more just a researchor's observations

Removing Whitelists and Staking (and SLAs?)

Problem:

Many kinds of activities require interaction between counterparties in which one needs the other to behave in a certain way. This is often done through the use of repeated games in the form of whitelists (if you misbehave now, I'll stop interacting with you) or through the threat of ex-post penalisation in the form of legal fines or slashing. All of these solutions can have drawbacks like capital inefficiency, overhead of monitoring and high barriers to entry (smaller set of possible counterparties), not to mention that they don't necessarily prevent bad behaviour.

The Sualution:

in some cases, we can encode good behaviour in a suave contract such that we can trust anyone running a kettle.

Examples:

- CoWswap solvers: currently CoWswap solvers have to place collateral to be solvers so that they can be slashed by CoWswap for misbehaviour. Running solvers in kettles with sufficient guardrails can prevent frontrunning risk and potentially other misbehaviour (currently unclear what the full list of possible misbehaviours is). This would remove the need for solvers to stake, allowing more solvers to enter the market as long as they run in kettles increasing competition and removing monitoring overhead.
- Block builders/relays: currently agents in the MEV ecosystem share order flow on whitelist-like trust. E.g. searchers will send OF only to builders they trust. This whitelist can be replaced or augmented by encrypting to all kettles.

Counterparty Assurances:

It's worth pointing out that the trusted party (not just the trusting

party as listed above) benefits from SUAVE-based trust. E.g. builders benefit from not having to convince searchers they're legit, solvers benefit from not needing to stake. This applies to almost anyone user facing like TG bots or CoWswap.

Enabling Richer Information Sharing

Problem:

Sometimes we see trusted relationships between different actors exist, but they exist in a throttled way due to limitations on trust. For example, searchers may be willing to send their transactions to other builders, but not more complicated programs that encapsulate more of their strategies even if this would cut down on latency as they don't want to leak alpha.

Solution:

As SUAVE can strengthen the guarantees one party can offer to another, more effective information sharing can take place (e.g. sharing more complex logic).

Cheap Verification

Problem:

when two parties do not trust each other, they must verify their claims. For Ethereum validators, this is done by re-executing all transactions to cross-check the state root. For ZK-rollups, this means computing proofs to be verified on L1/by full nodes. These techniques for verification can be expensive.

Solution:

TEE signatures provide a cheap form of proof (signatures). The trust assumptions are weaker, but this is OK in many cases.

Examples:

- TEE-rollups for lower-security operations could roll-up to suave or other chains
- Relays in kettles can assert block validity and payment of bid in a block