Invoice - based reputation system

Jan Betley
Golem Factory

March 14, 2022

1 The main concept

General note: this is a crude description of an idea, not a "final" design I'm covinced will work. IOW, this document should be enough for discussions and future plans, not something we can now go and implement.

1.1 Assumptions

The most important assumption in this document is the expected utility maximization hypothesis[1]: providers and requestors will do what they think is best for them, and the main purpose of the reputation system is to construct an environment where the most profitable behaviour is also the desired behaviour.

Few more assumptions (some of them are questionable even as approximations - there is a further section discussing them):

1. The ground truth about all payments on the market (timestamp, amount, sender, receiver) is known.
2. The ground truth about all invoices on the market (timestamp, amount, sender, receiver) is known.
3. Obtaining & parsing the data about invoices/payments costs neither money nor time.
4. The only thing provider cares about is "how much I will be paid?" and the only thing requestor cares about are the quality of the services received and amount paid.
5. The market is balanced, i.e.:
   (a) Good enough and sufficiently cheap provider will spend no more than X% of time waiting for agreements, for a reasonable X (e.g. 10%).
   (b) Requestor willing to pay a sufficient amount of money will find enough providers fast enough

[1] https://en.wikipedia.org/wiki/Expected_utility_hypothesis

1.2 General idea

The full knowledge about invoices and payments gives us a crude approximation of the "reputation-like" information: good requestors pay, good providers are paid. In more detail:

1. Provider POV: the requestor who paid bigger part of their past invoices will more likely pay another invoice than another requestor who paid less past invoices.
2. Requestor POV: the provider whose invoices are usually paid more likely provides high quality services than a provider who is paid only rarely or never.

Justification behind these statements:

- Both first and second: providers/requestors market strategy rarely changes, so it's likely that when trading with us they will behave similarly to the way they behaved in the similar past situations.
- Second: if a requestor doesn't pay the provider, they are less likely to trade with them in the future (e.g. because of the provider's local history), and thus requestors more often pay providers they want to trade with, and those are more likely the providers we want to trade with.

1.3 Detailed strategies

There are countless possible strategies that are consistent with the main idea, strategies in this section are just examples.

Provider strategy:

1. Gather a local history of all our contracts - (requestor id, invoice amount, paid amount) + additional details (e.g. agreement length, initial demand etc).
2. Send offers to the market with "base" prices, that will later (in the negotiation phase) be increased for untrusted requestors.
3. When a requestor responds to the our initial offer:
    - If this is a new requestor (i.e. one that didn't yet trade with anyone on the market, or didn't trade enough):
        (a) Counteroffer with a "payment in advance" property
        (b) If accepted, start with sending a debit note (for some fixed amount) with a short payment time
        (c) Refuse to start an activity until the first debit note is paid
        (d) Continue in the same manner, i.e. request upfront payments
    - If this is a requestor we've already traded with, and they didn't pay all the due amount:

(a) Add the missing amount (possibly with some discounting factor) to the fixed amount
(b) If agreed, start with sending a debit note for the missing amount, with short payment time
(c) Refuse to start an activity until the debit note is paid

- In other cases, i.e. when requestor has some experience on the market and they either didn't trade with us or paid a full amount:
  (a) Estimate the % of the final invoice we expect the requestor to pay us. In the most primitive version this might be just the % of their invoices they paid until now, although this can (and should) be done much better[2]
  (b) If we estimate ∼100% of the invoice will be paid, accept the offer.
  (c) If we estimate less will be paid, reply with a counter-offer with higher prices, so that they compensate for the % of the invoice we expect not to be paid.

Requestor strategy:

1. Gather similar local history as the provider
2. Sign agreements that require an advance payment only with providers you have a reason to trust (e.g. because you traded with them before or because they have a good history on the market)
3. Except for the previous point, just try to sign the most profitable agreements, including any information about the "provider quality" you have. This should include "high quality providers are paid more often than low quality providers", but also other things like "provider benchmarks" or the history of the requestor's trades.
4. Provider sends debit notes/invoices stating the expected amounts, but requestor decides freely about the amount paid (they might also decide not to pay at all). When deciding about the amount to pay, consider a following tradeof. The less money I pay:
   - (gain) The more money I still own.
   - (loss) The more expensive deals I will have with other providers, especially if I decide not to pay a provider who's usually paid.
   - (loss) The more I will have to pay to trade with this provider again.

[2] E.g. if they didn't pay a provider that is never paid, there's a big chance this is an invalid provider who just does no work at all, but if they didn't pay a provider who's usually paid fully then probably something's wrong with the requestora. We should also include in the estimations few other types of the information, e.g.

- Our own experiences with this requestor (i.e. requestor who dutifully paid us a lot will probably pay again).
- Expected agreement length, proposed payment interval etc.

And pay any amount you consider best from your POV.[3]

1.4 The reputation system deployment process

So we have defined some strategies for requestors/providers. Who will use them? How to encourage others to use similar strategies? The plan looks as follows:

1. Implement provider/requestor strategies described above. Set them as default provider strategies and recommended requestor strategies.
2. Put on the market enough providers/requestors who use these strategies.
3. (Crucial assumption that should be analyzed) When a new provider/requestor enters the market, they use these (or similar) strategies, because they are the most profitable. Thus, as the market grows, the adoption level of these strategies remains high.
4. Finally, we can remove the initial providers/requestors and the reputation system will still work.

2 Discussion

2.1 Assumption 2 - ground truth about invoices

This is hard, and this must be solved well for this whole idea to work. There are two general directions:

1. Implement some logic that will enforce this ground truth, e.g.:
   - Put invoices on some additional chain
   - Require invoices to be signed by both sides (?)
2. Drop this assumption, but modify the strategies so that "agreeing on the invoice amount" will be profitable for both sides, e.g.
   - Both provider and requestor "report" invoices somewhere
   - For the requestor it's always (or usually?) better to report the same value that was reported by the provider
   - For the provider it's always (or usually?) better to have the invoice value confirmed by the requestor

[3] This should somehow converge to "the worse is the provider, the less we want to pay" - and that's exactly what we want.

2.2 Assumption 3 - obtaining/parsing invoice data cost

We'll need a node information API (GAP-14) that will gather the data and provide useful simple aggregates that will be consumed in strategies.

This looks like a fully centralized solution, but:

- If the invoices are gathered in a public location (e.g. a side chain) then there can be multiple different APIs aggregating the same information.
- If the invoices are gathered in a database without a public access, then there could be some other entity who also gathers invoice data.

2.3 Other assumptions

- Assumption 1 (payments ground truth) is true, because we pay in crypto.
- Assumption 4 (providers care only about money, requestors care only about service quality) is false[4], but this is simply what this reputation system gives us - and it's quite a lot, hopefully enough for a "serious" start of the market.
- Assumption 5 is fulfilled for any functioning market.

2.4 Misc random notes

1. There is a malicious strategy available:
   (a) Put my own providers and requestors on the market
   (b) Trade between them only until their reputation grows enough
   (c) Use requestors to steal from other providers, use providers to steal from other requestors

   This is an important problem, but hopefully solvable:
   - Because of the transaction costs, building up reputation this way costs
   - Provider-requestor clusters like this will be visible in the data. E.g. a provider who got paid by X different requestors surely looks better than a provider who was paid multiple times by only one requestor.
2. Maybe debit notes would be better than invoices? This seems to be a good direction, because it gives us an additional useful information - the amount accepted by the requestor. Accepting and not paying later is much worse than just rejecting the debit note.
3. The obvious vulnerability is that if an estabilished requestor (i.e. not paying in advance) intends to leave the market, then there's no incentive for them to pay the last invoice(s). This is true, but not really important - estabilished requestors will be rarely deciding to leave the market.

[4] e.g. requestor would also want their data not to be stolen.