Monetha

MONETHA PLATFORM

Scoring Layer

What it is

Reputation scoring is widely used in many domains to measure the trustworthiness of a target (a user or service) based on information about it and its past behavior. Users observe the events that immediately followed the target's behavior, which determines how they perceive the experience. Events depend on a context and can be, for instance, voiced opinions, transactions, or signed documents. Thus, the reputation of a target is a collective measure of trustworthiness built from users' experiences. With reputation systems, users can rate the behavior other users, expressing their direct judgements.

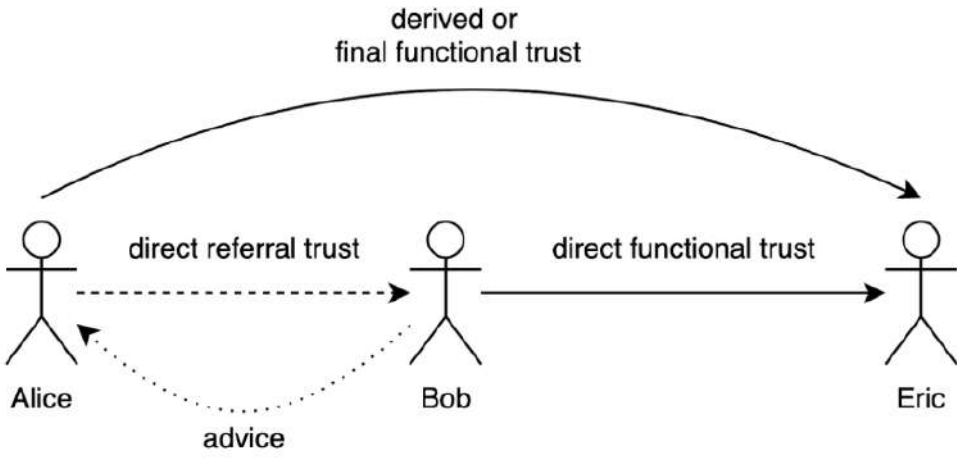
Our Scoring layer aims to calculate the resulting reputation score by collecting, distributing and aggregating the pieces of evidence (events) about a target's behavior. As a result, before engaging in an interaction with a target, the user can consider the reputation of the target and start interacting with it only if it is trustworthy. Thus, reputation gives the extent to which the target's behavior is good or bad.

Monetha's platform uses several models to aggregate trust information and compute a score, namely flow-based reputation models and Subjective Logic. Flow-based reputation models provide an automated method for aggregating all available trust information. Subjective Logic extends these models with an explicit notion of uncertainty, incorporating a margin of error into score calculation due to the (limited) amount of available trust information.

Captured events, which in the context of some proposition can be either positive or negative, are used to build the user's opinion about the proposition — the central concept in Subjective Logic. An opinion about some proposition consists of three parts: belief, disbelief, and uncertainty. Belief represents that a proposition is provable, disbelief represents that a proposition is disprovable, and uncertainty represents a lack of evidence, or the belief that a proposition is neither provable nor disprovable.

Subjective Logic uses a consensus operator, which is rooted in the theory of evidence, to fuse independent opinions. It also uses a discounting operator, which is based on a probabilistic interpretation of opinions and represents the flow of evidence from one party to another, to compute trust transitivity. During the flow of evidence, a lack of trust in the party that provides evidence is translated into a reduction of the amount of evidence. That makes it a suitable mathematical framework for handling trust relations.

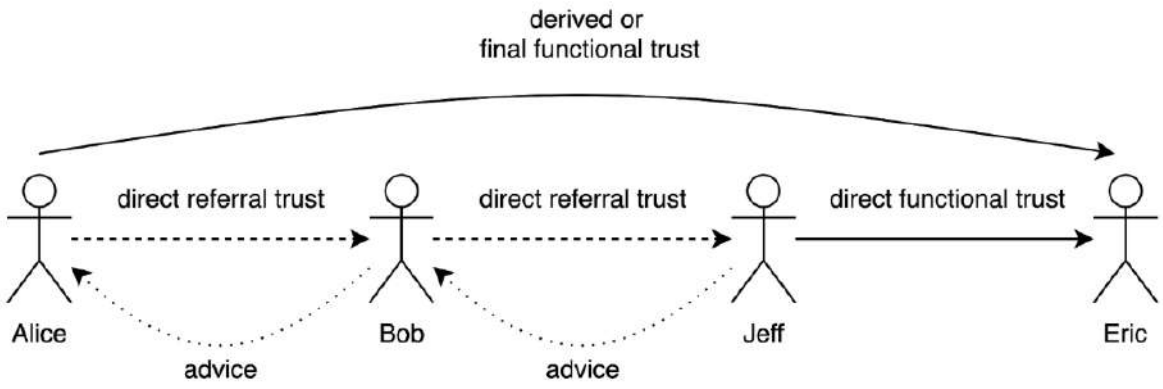
As an example, suppose Alice needs to buy a car. Assume further that Alice has no experience with purchasing cars. Bob, who is one of her colleagues at work, has already bought a used car from a dealer, and was very pleased with the purchase. Bob gives her a lift with his car. Alice notices that the car was in very good condition and did not need any repairs for a long time. So she intuitively trusts Bob in matters of car purchasing. Bob tells her that this is the second used car in excellent condition he has bought from a dealer named Eric. Based on his direct experience, Eric seems to be a very skilled car dealer. As a result, Bob has direct trust in Eric and advises Alice to buy a used car from him. Based on her trust in Bob in the matters of car purchasing, and on Bob's advice, Alice develops trust in Eric too. Alice's newly derived trust in Eric is indirect because it is not based on direct experience.



This example represents trust transitivity, in the sense that Alice trusts Bob who trusts Eric, so that Alice also trusts Eric. This assumes that Bob actually tells Alice that he trusts Eric, which from Alice's perspective is opinionated advice or a recommendation.

In Subjective Logic, trust is represented as an opinion or as a tuple which consists of belief, disbelief and uncertainty. That means Alice uses her direct referral trust in Bob to derive or compute final functional trust in Eric from a subjective trust network by applying the trust-discounting operator on the Bob's direct functional trust in Eric.

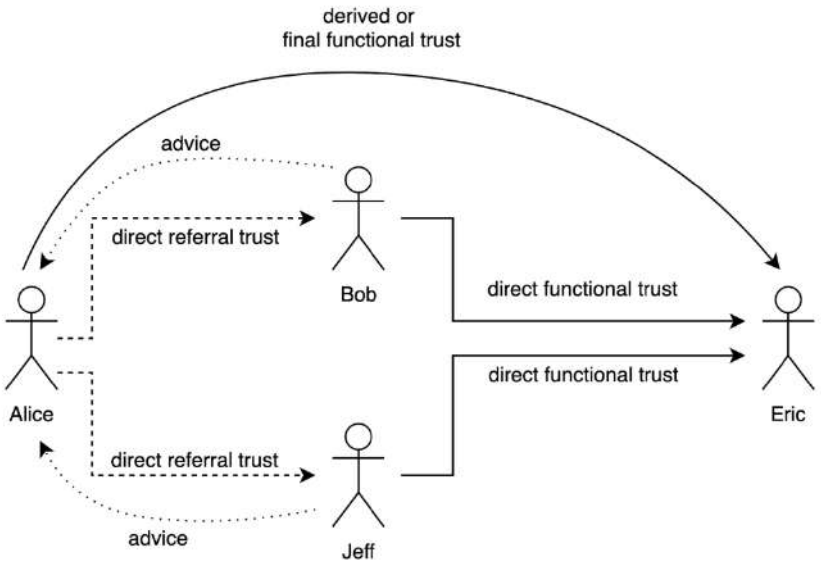
Let us slightly extend the example. In this scenario, Bob does not actually know any car dealer himself. However, he trusts Jeff, whom he believes knows a good car dealer. As it happens, Jeff is happy to give positive advice about Eric to Bob, which Bob passes on as advice to Alice. As a result of transitivity, Alice is able to derive trust in Eric, as illustrated below:



In terms of Subjective Logic, Jeff's direct functional trust in Eric is discounted using Bob's direct referral trust in Jeff, then computed Bob's final functional trust in Eric is discounted using Alice's direct referral trust in Bob to get Alice's final functional trust in Eric. This also illustrates a subtle difference between functional and referral trust. Referral trust in this example is about the ability to give advice about a dealer who can sell a used car in good condition. Functional trust is about the ability to actually sell a used car in good condition.

It is common to collect information from several sources in order to be better informed, such as when making decisions. In case of opinions, this can be called trust fusion, meaning that derived opinions resulting from separate trust paths are fused into one.

Let us continue the example of Alice who needs to buy a car, where she has received advice from Bob to buy the car from dealer Eric. This time we assume that Alice has doubts about Bob's advice, so she would like to get a second opinion. She therefore asks her other colleague Jeff for his opinion about Eric. The trust graph which includes both pieces of advice is illustrated below:



In this example, the trust opinions derived from each path are first computed with the trust-discounting operator, then the two derived trust opinions are fused with the consensus operator.

This trust fusion example uses a combination of trust discounting and fusion. By combining fusion and trust discounting, complex trust networks can be modelled and analysed. Flow-based reputation models provide an automated method for computing reputation scores for arbitrary trust networks.