This is work done with Stephen Clarke, who’s just there lurking, and Hannan Xiao, who was here yesterday but can’t be here today. The motivation is that we often want to do business, or conduct transactions, with strangers, people we haven’t done business with before. Reputation systems don’t really work, or at least reputation systems for giving reputations to strangers don’t work. Now most people say that that’s because you don’t know the people giving the reputations, I want to make the slightly stronger claim that even if you did know all the people who gave all the reputations, and you trusted them all, that still wouldn’t help. This is basically because trust isn’t transitive. The fact that Alice trusts Bob, and that Bob trusts Carol, isn’t enough to ensure that it’s appropriate for Alice to trust Carol, because the fact that Carol gives Bob a good service doesn’t mean that Carol is going to give Alice a good service. It might be that Bob is a regular customer, it might be that there’s some other reason why Carol’s giving Bob a good service. Perhaps she fancies him. The assumption that we’re going to make, for the purpose of this talk, is that local trust management is a more tractable problem than the one that we started with here. If you have people that you do business with all the time, then there are various systems that more or less allow you to do get a particular service in such a way that it’s either cheap or reliable depending on which has more utility in that circumstance.

So what we wanted was something that would allow us to reduce the “global trust management of strangers” problem to a “local trust management for people who regularly do business with each other” problem, that we thought might be a bit more tractable. But we didn’t want to reduce it to a particular local trust management mechanism, we wanted something that would allow you to use whatever local trust management system you prefer.

In the real world there’s quite a straightforward solution to this which is the use of third party guarantors. If you want to rent a flat from a landlord and he says, ah you’re a student, you have to go and find someone willing to guarantee that you will pay the rent. The point is that that doesn’t make the landlord trust the tenant any more than they did, OK, there’s still no transitive trust, it’s simply that if Alice doesn’t pay the rent, Carol trusts Bob to make her whole. Bob then has the problem of whether he was right to trust Alice, and whether Alice is eventually going to repay him. Even when it all works, people’s assessments of the probability of something going wrong, or the cost of something going wrong, are likely to be very different. It may simply be that Bob has more knowledge.
than Carol and consequently he’s more willing, because he believes he’s taking a much smaller risk than the risk that Carol believes she’s taking, and perhaps he’s willing to do it in exchange for some compensation. What’s Bob’s incentive? Maybe Alice pays him commission, maybe Carol pays him commission, possibly out of the rent that Alice pays when she pays. But the key point is, that all the trust relationships are now local trust relationships between people who know each other and do business with each other. Clearly each trust relationship can get updated independently. If Alice defaults, either Bob pays Carol or he doesn’t, either Alice pays Bob or she doesn’t, each trust relationship is being updated locally, independently, on the basis of a local transaction. And clearly you can extend this idea to multiple hops, and that’s the relationship that we rather naughtily call Trust*.

So here you have the idea of David trusting Carol, Carol trusts Bob, Bob trusts Alice, there’s some notion of a forfeit; there’s a notion of a guarantee with a forfeit being paid if the guarantee is broken. For the moment think of the forfeit being a micropayment, using whatever your favourite micro payment system is. We started off using keynote because we could do both the local trust management and the micro payment in keynote. But the micro payments are also all local, you never make a payment to somebody who is more than one hop away from you in the chain.

Now it’s important to notice that even in just a one hop relationship, Trust* isn’t quite the same as trust, because with Trust* what I am asking Bob to believe is that Alice will either provide the service or pay the forfeit; and Bob might believe that Alice is completely hopeless at providing the service and will never succeed in doing so, but also believe that she will give him a penny every time she doesn’t, so he will just sit there clicking until Alice gets tired and stops paying, or withdraws the service.

**Ben Laurie:** I don’t quite understand how this maps to the example you gave before, where Alice is the tenant and Bob is the guarantor, what service is Alice providing Bob when Alice is the tenant? And what forfeits does Alice pay? I’m not understanding how these two map onto each other.

**Reply:** In this example here Carol is the landlord, Alice is the tenant. Let’s assume for the moment that Alice pays Bob a commission to guarantee her, Alice pays Bob a certain proportion of the rent to act as a guarantor, and that’s the service that Bob is providing to Alice. The service that Bob is providing to Carol is that if Alice defaults Bob will then pay the rent.

**Ben Laurie:** Yes, and what’s the forfeit for Alice?

**Reply:** That’s between her and Bob. It may be that the forfeit that Alice has to pay Bob is bigger than the forfeit that Bob has to pay Carol, in which case Bob will be hoping that Alice fails. It may be that Alice pays Bob a flat commission, and in that case Bob is hoping that Alice doesn’t default because he’ll be out of pocket if she does. There are some interesting consequences to those trade-offs, which I’ll talk a little bit more about later.