

A
alueflows
story about
Pie

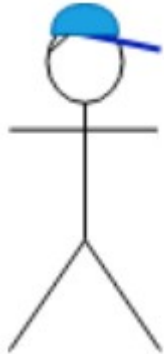


Part 1: Making Pies

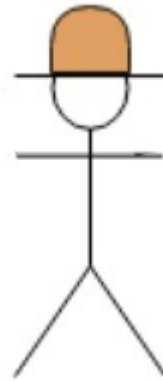
Alice, Bill, and Carol are *agents*. An agent can be a *person* or an *organization* (formal or informal). For this example we just have people, but you can imagine organizations doing the same things.



Alice (a baker)



Bill (has a truck)

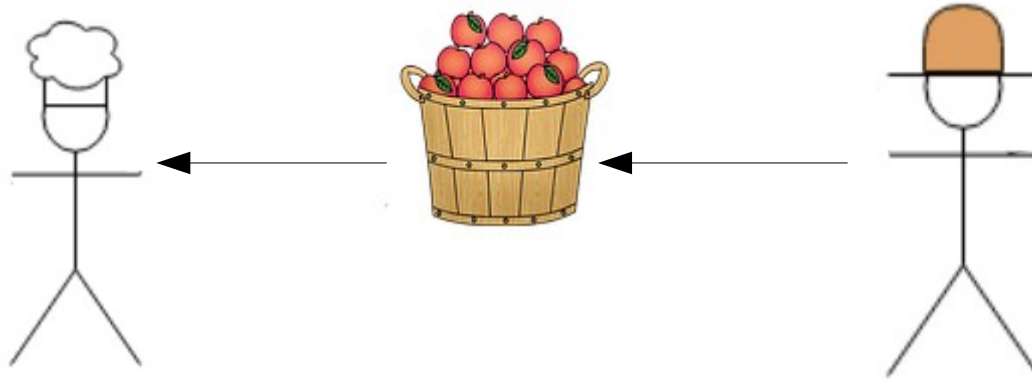


Carol (an apple grower)

Alice bakes pies. Pies are *economic resources*, yum.

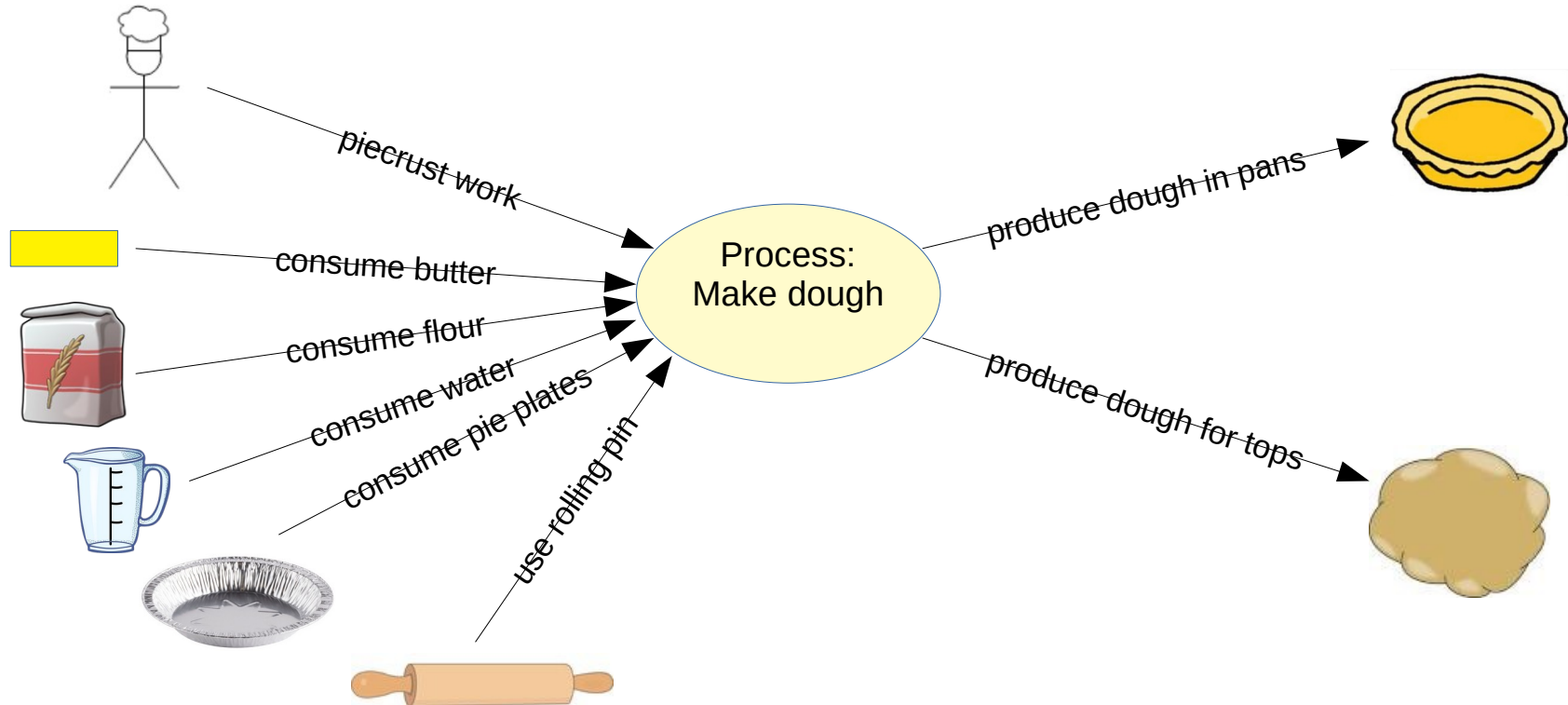


So to start our story, Alice gets apples from Carol. She has promised Carol one pie per half-bushel of apples. This is the first *transfer* of that *exchange*.

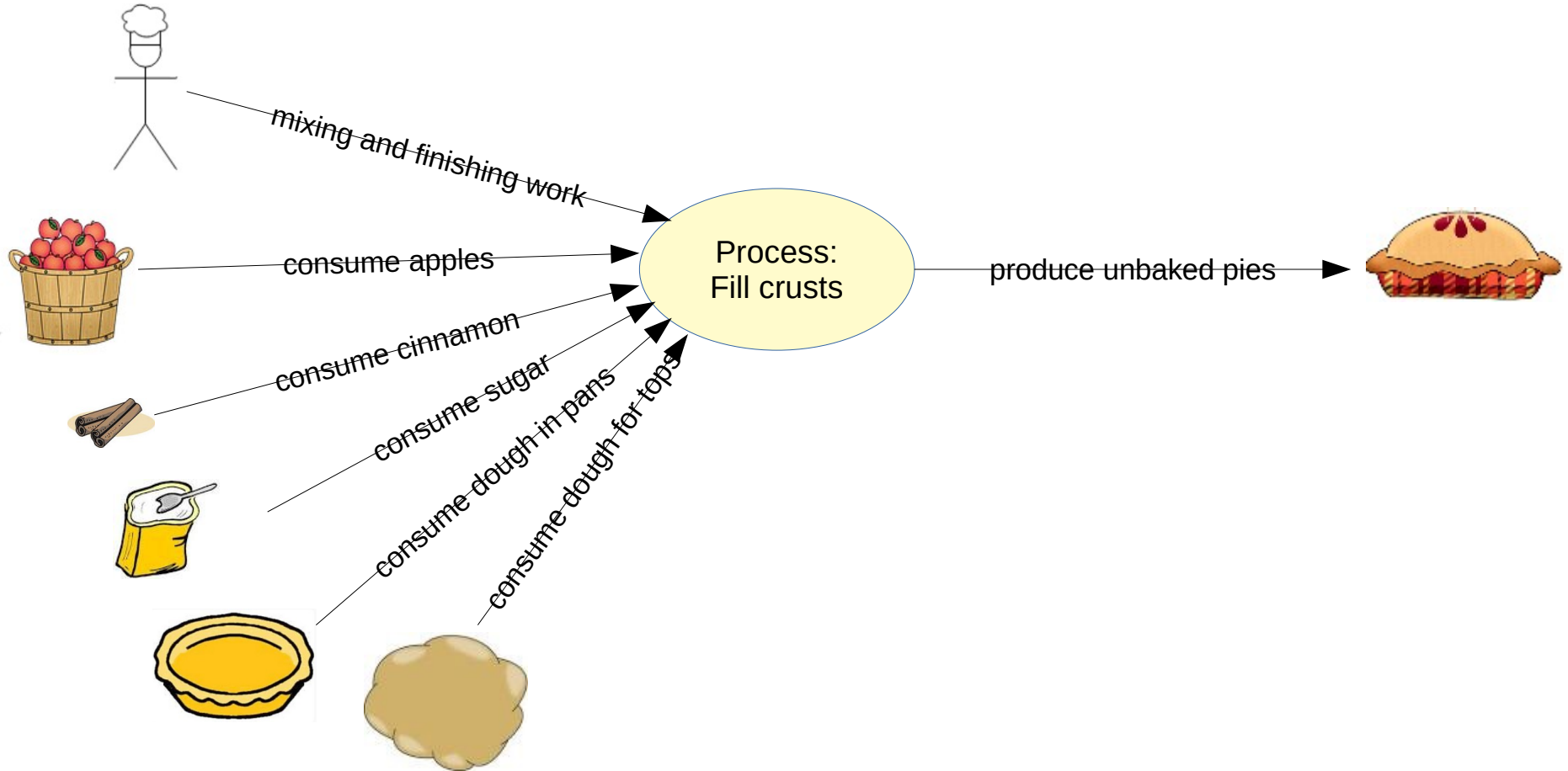


Alice makes her pies in 3 steps. Each step is a *process* with input and output *economic events* (the lines) that can affect *economic resources*.

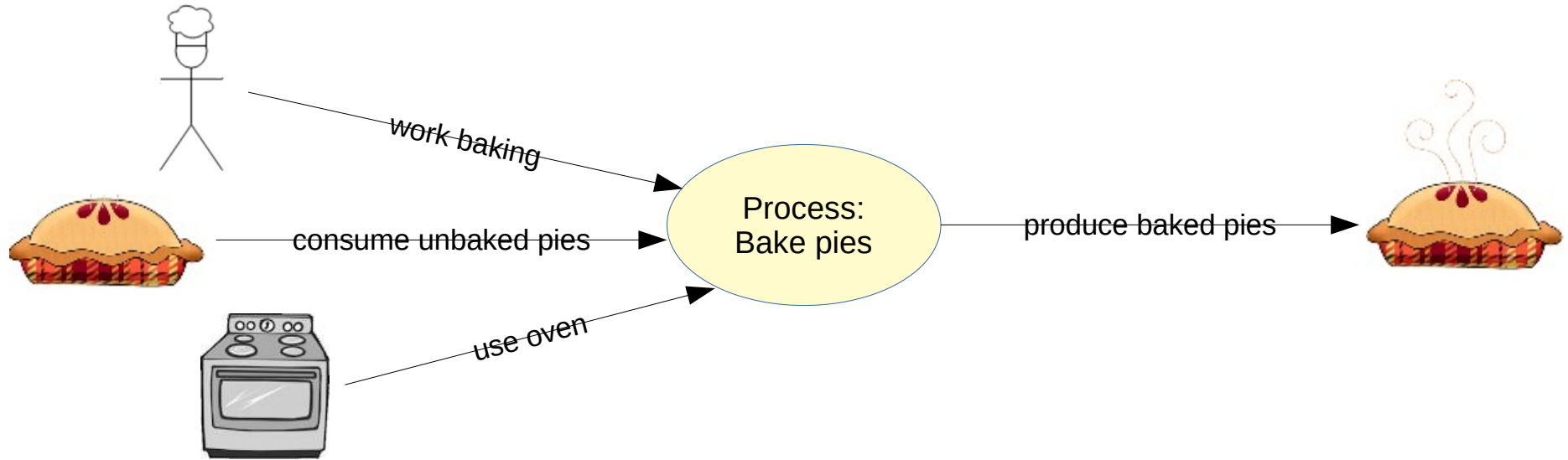
First, she makes the crust.



Then she makes the filling and fills the crusts.

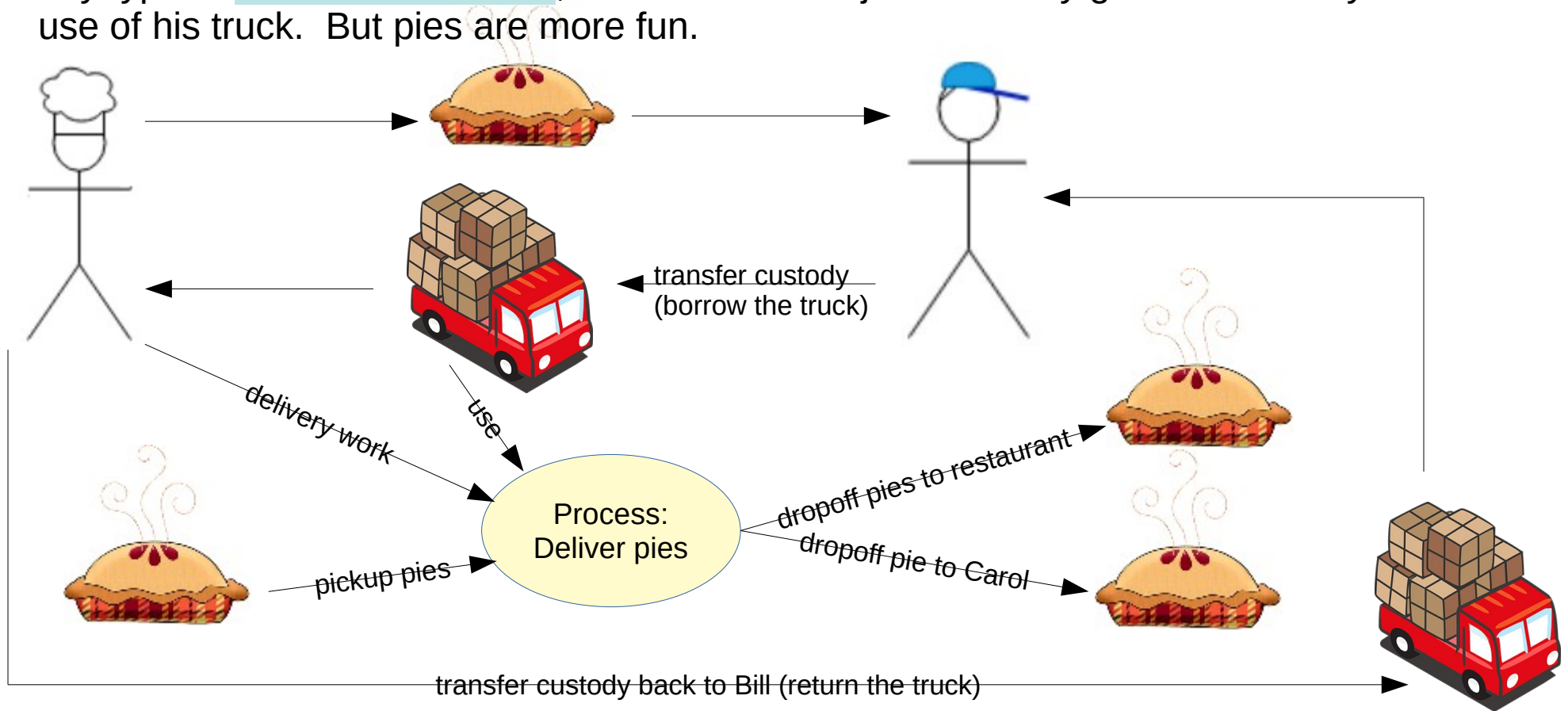


Then she bakes the pies.



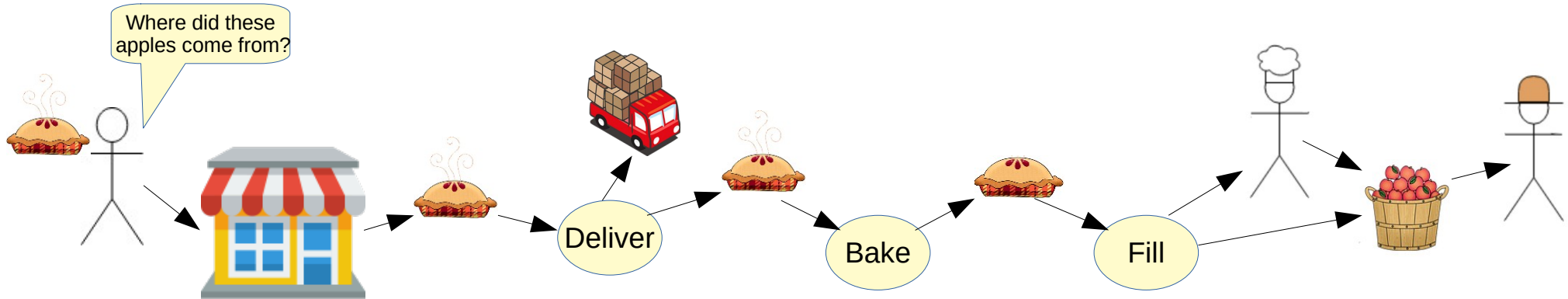
Maybe you are wondering why Alice chose to configure this as 3 processes. Well, it gives her a lot of flexibility. For example, she can make a lot of dough and use it for blueberry and pumpkin pie, as well as apple pie. Then she might want to bake different kinds of pies in one oven at the same time too, or freeze half of the unbaked apple pies to bake later.

Alice borrows Bill's truck to deliver her pies to a restaurant that made a special order. In return for the use of his truck, she gives Bill a pie. Note that Valueflows handles any type of *economic resource*, Alice could have just as easily given Bill money for use of his truck. But pies are more fun.



So now Alice has made and delivered her pies, creating flows of value. We can follow the flows forwards and backwards. What if the restaurant that has bought some of Alice's pies wants to tell its customers about where the pies come from?

If they trace the flows backwards, they might find out that the pies are made by Alice, who is a local baker and they can see that she uses all organic ingredients. They might find out that Alice gets her apples from Carol, who is a cooperative worker-owned apple orchard. That all makes the restaurant customers even more hungry for apple pie....



Part 2: Planning

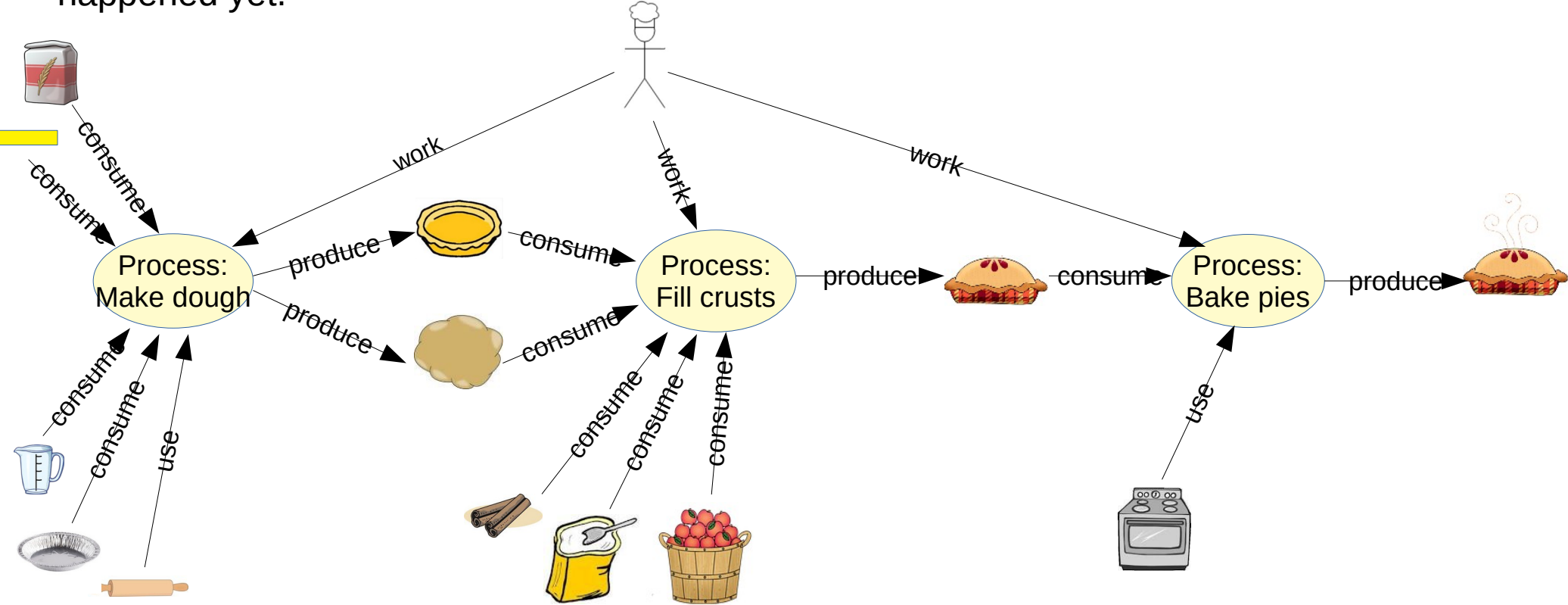
Alice needs to schedule her bakery work so everything she needs gets made, all the ingredients are available when she needs them, the equipment is all available for all the different things she creates, etc. In short, she needs a *plan*.

To keep it simple, let's check in on Alice as she plans to make the apple pies from Part 1.

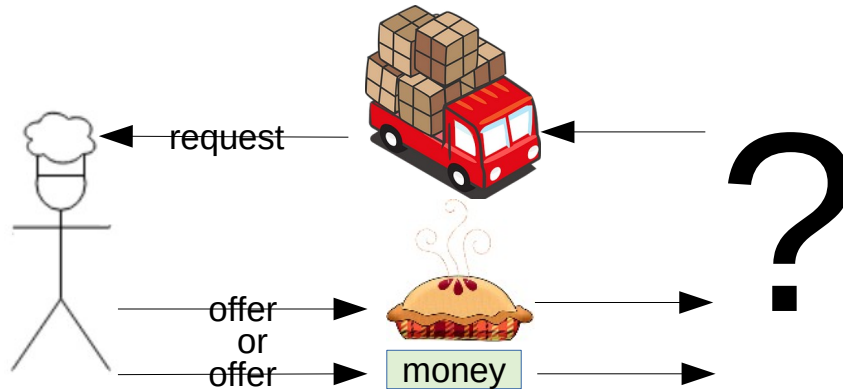
Planning starts with what you want, the *independent demand*. The restaurant wants 20 apple pies by Thursday 3pm. Alice also wants 8 apple pies to have for sale in the bakery, plus one for Bill and one for Carol. So her total independent demand for the plan for Thursday is 30 apple pies.



Alice's plan looks something like this. Of course, this is simplified, for example there would be quantities and scheduled times on all of these lines. On a *plan*, the lines are *commitments* rather than *economic events*, although they look pretty much the same, which is by design. *Commitments* are promises for *economic events* that have not happened yet.

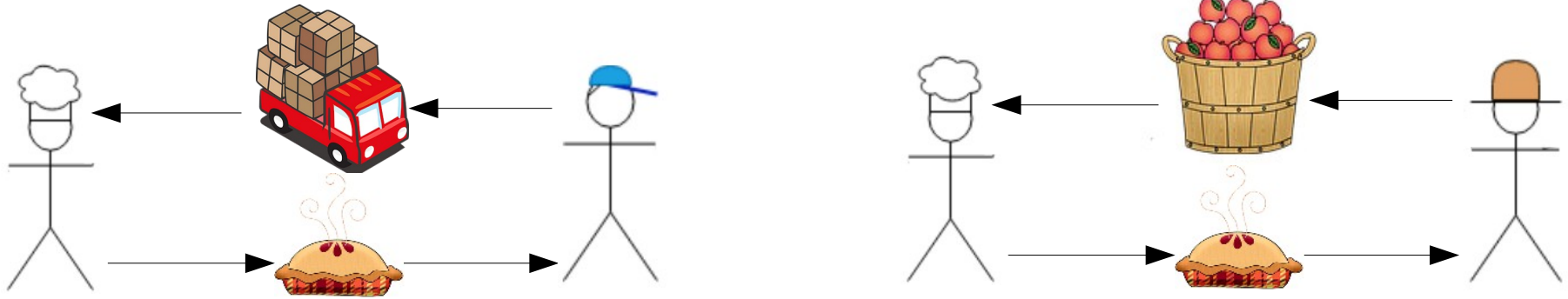


Alice already has a standing *agreement* with Carol to trade apples for pies, because Carol gives away pieces of pie to customers who pick their own apples. But the restaurant pies are a special order, which Alice promised to deliver. And Alice doesn't have a truck, so she publishes a *proposal*, which has *intents* instead of *commitments*. *Intents* are looking for matches so there can be some commitments.



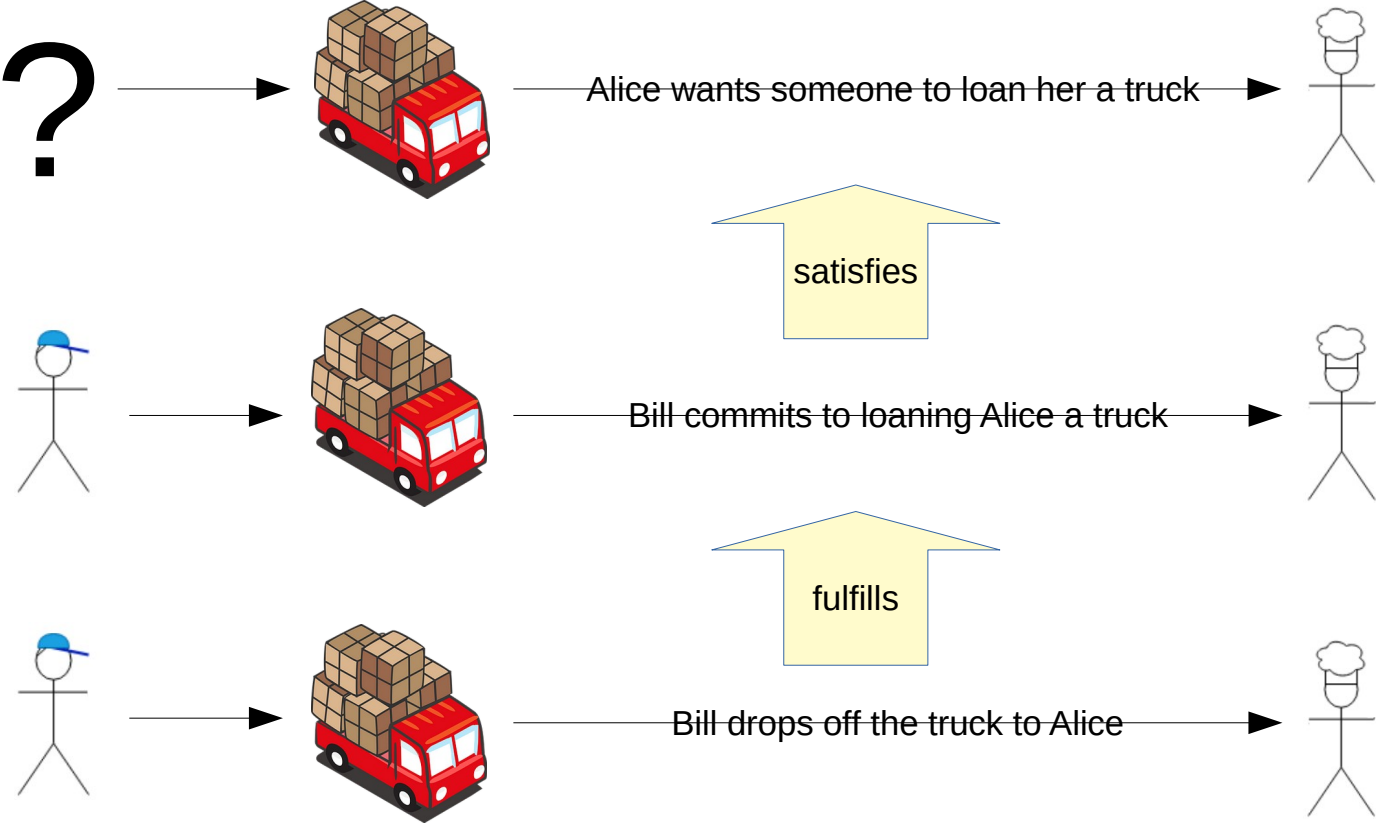
This is Alice's *proposal*. The lines are the *intents*, and the only *agent* we know is Alice.

Bill finds Alice's *proposal* and agrees to loan her his truck. Bill wanted the pie in return, who wouldn't? Alice also didn't have enough apples onhand, so she arranged with Carol to get some more. Now she has the *agreements* she needs.



The lines here are *commitments*. The pies and apples are not real yet either, so are just *resource specifications*. Bill committed use of the actual specific truck though, it is an *economic resource*.

As you might expect, when the *economic events* are recorded, they explicitly fulfill *commitments*, if planning was done. And if someone commits to an *intent*, the *commitment* explicitly satisfies that *intent*.



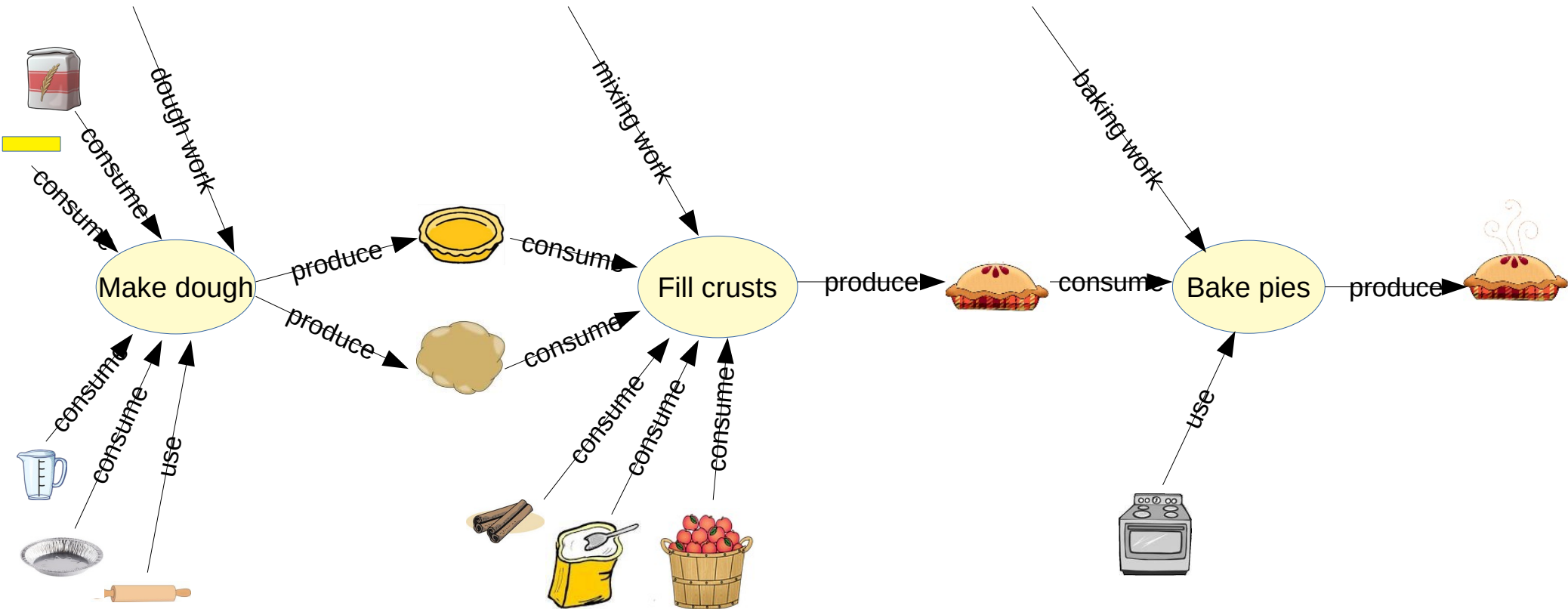
Part 3: The recipe

But Alice usually makes the same kinds of things over and over. Planning those each time takes a lot of time.... unless you have *recipes* that can generate *plans* for you. We all know about *recipes* for pies, but we are using that term here to include anything you can make, provide as a service, and really anything you do more-or-less the same way each time.

Software that generates plans from recipes can actually be pretty smart. It can check for what Alice already has in inventory and schedule only the remainder needed, for example. It can tell Alice what inputs she doesn't have enough of and can either generate a plan from a recipe that makes those or schedule trading for those ingredients.

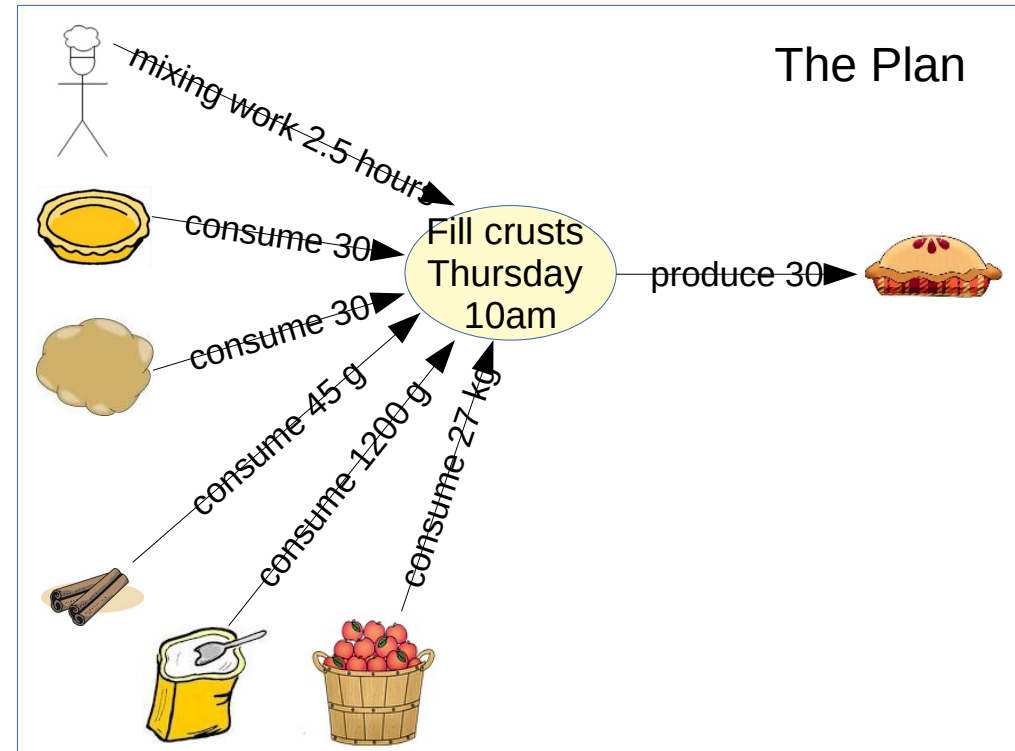
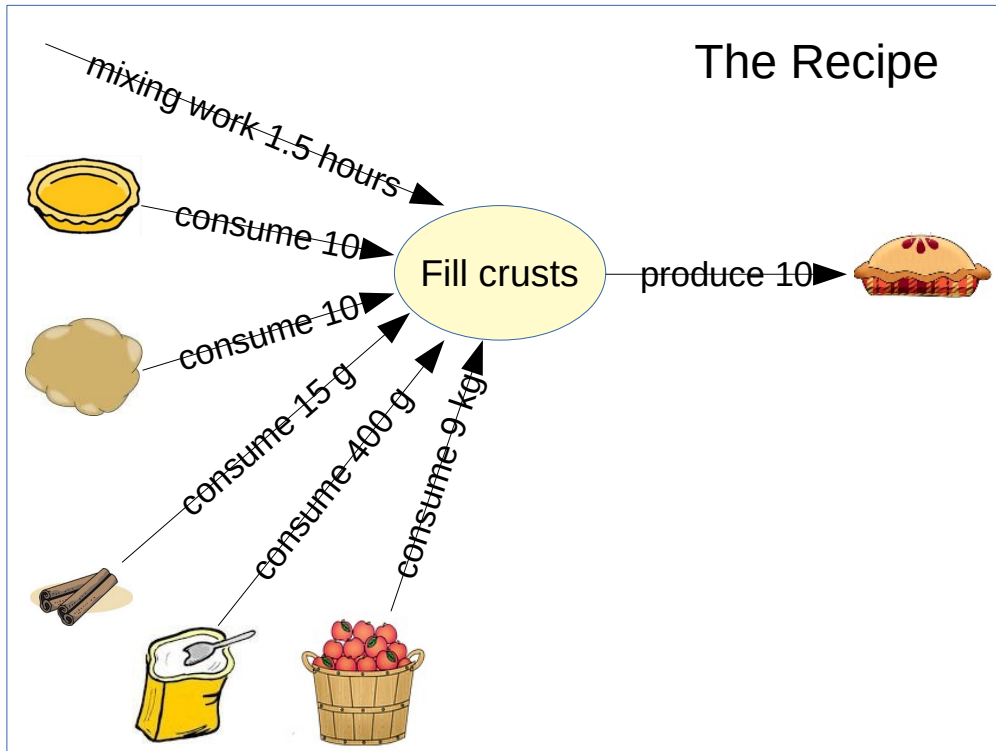


Not surprisingly, the *recipe* looks a lot like the *plan*. Except there aren't any specific *economic resources*, only *resource specifications*. And there aren't *processes*, only *process specifications*. And there aren't any real *agents* assigned, only the types of work that is needed.

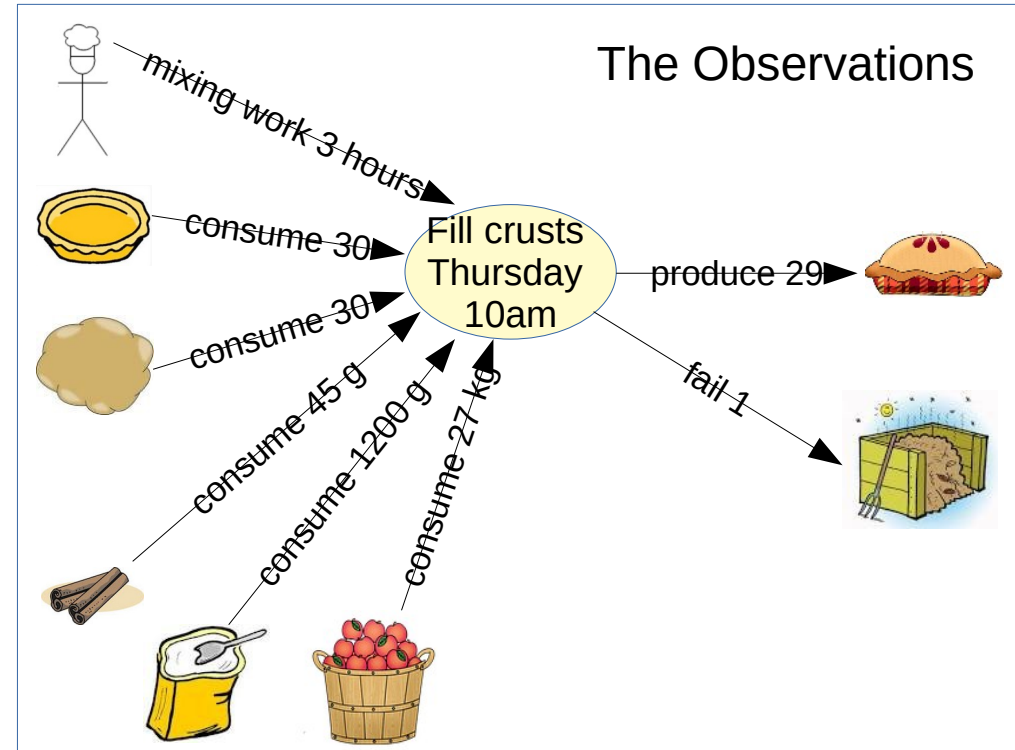
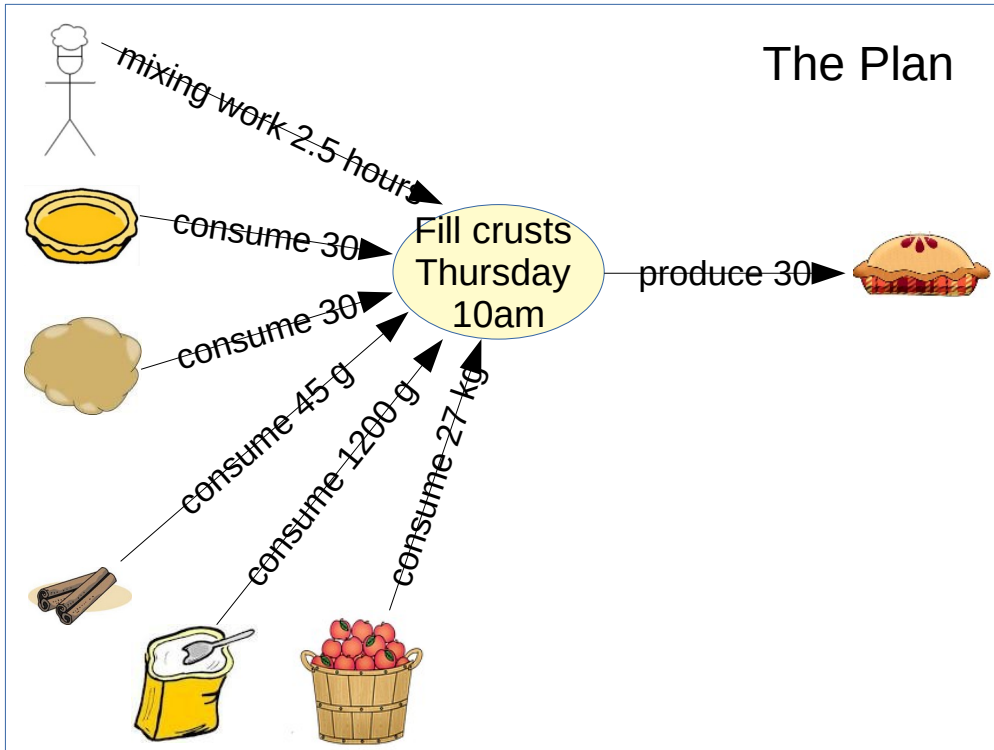


Some other differences between the *recipe* and the *plan*:

- The *quantities* are often different because the quantities in the recipe are adjusted to fit what the plan needs to output in the end
- The recipe has no *dates* or *times*, while the plan is scheduled with specific dates and times



Just to complete the circle of this story, here is what the plan vs the observable making of the pies might look like. It took a little longer than planned because Alice dropped a pie, which went into the compost.



Thank you for listening to the story of the apple pies. We hope this has helped you to understand Valueflows and the REA (Resource-Event-Agent) model it is based on.



Image attribution:

<https://classroomclipart.com>

<http://www.iconarchive.com>

<https://pixabay.com>

<https://www.flaticon.com/>

<https://findicons.com>

<http://clipart-library.com>