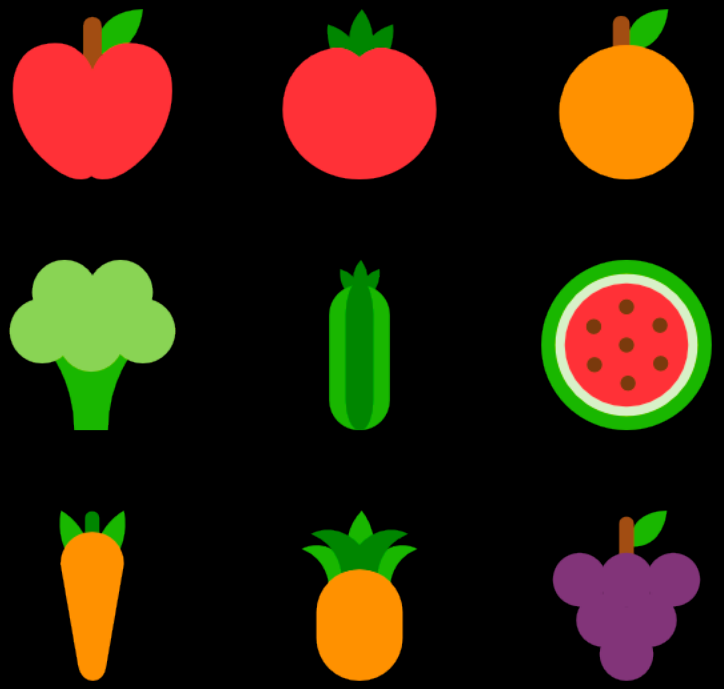
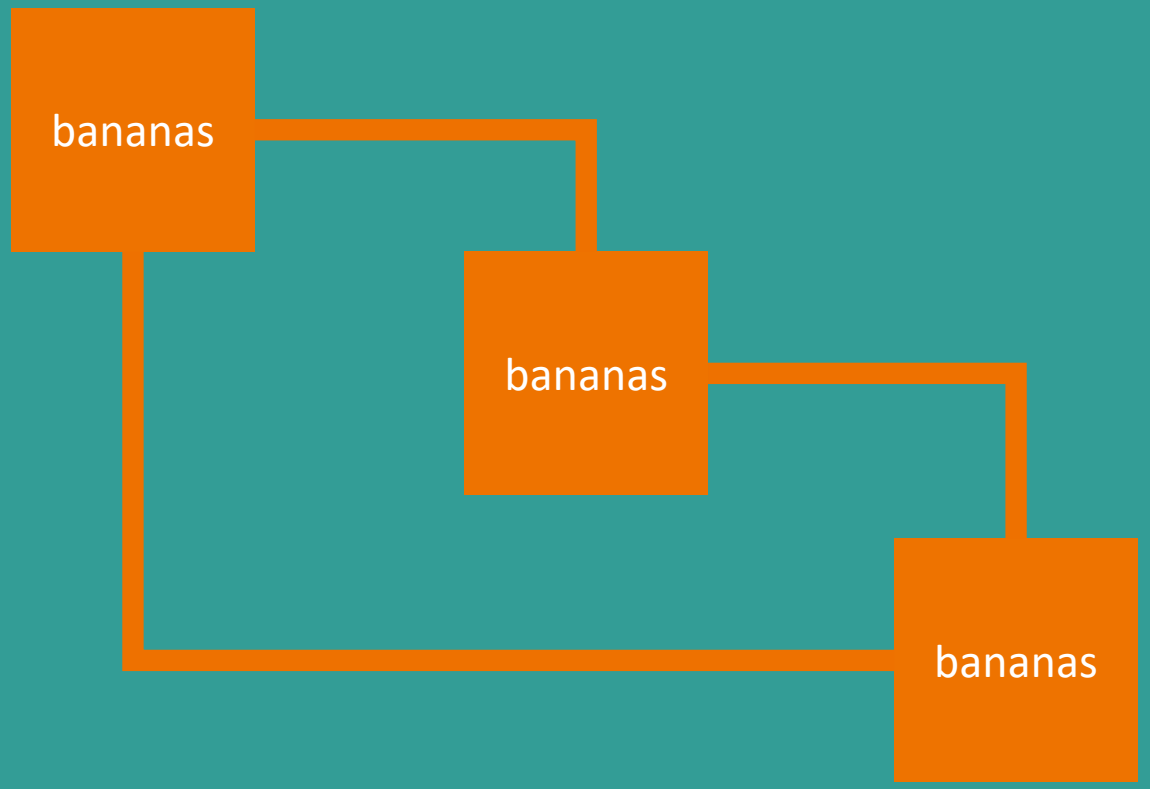
DIGITAL INNOVATION

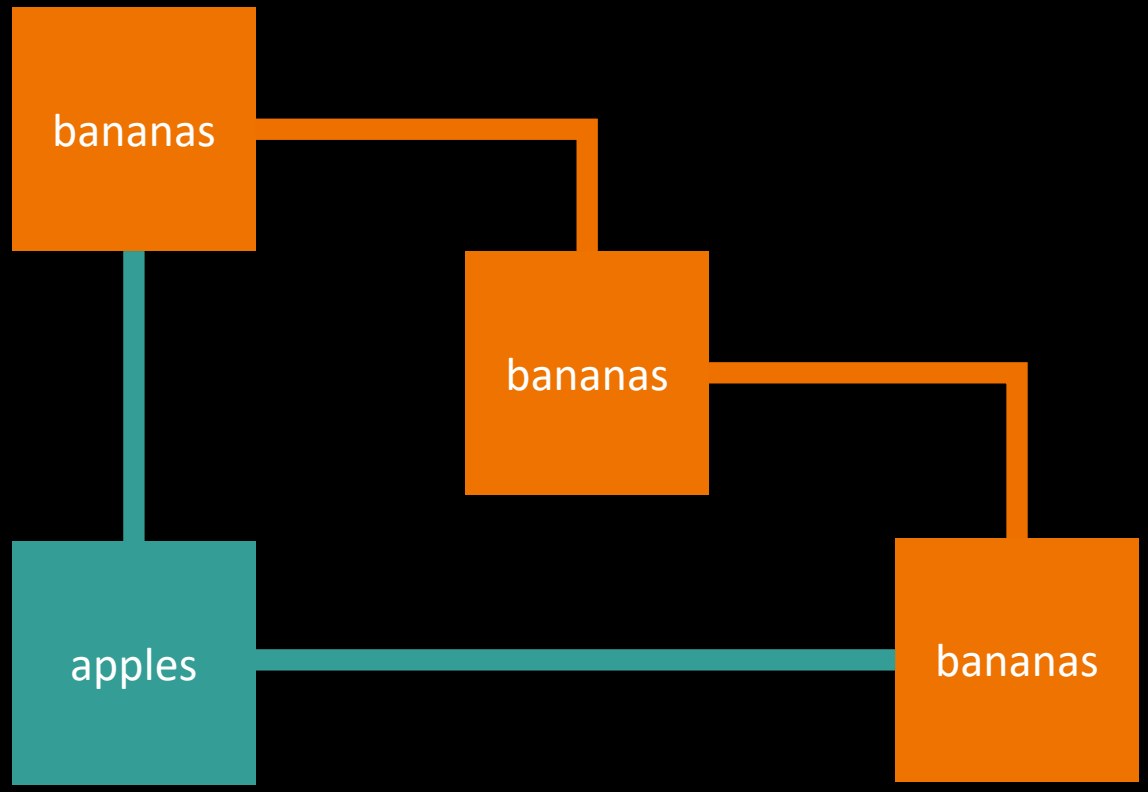
Service-Orientation

A fruit market analogy

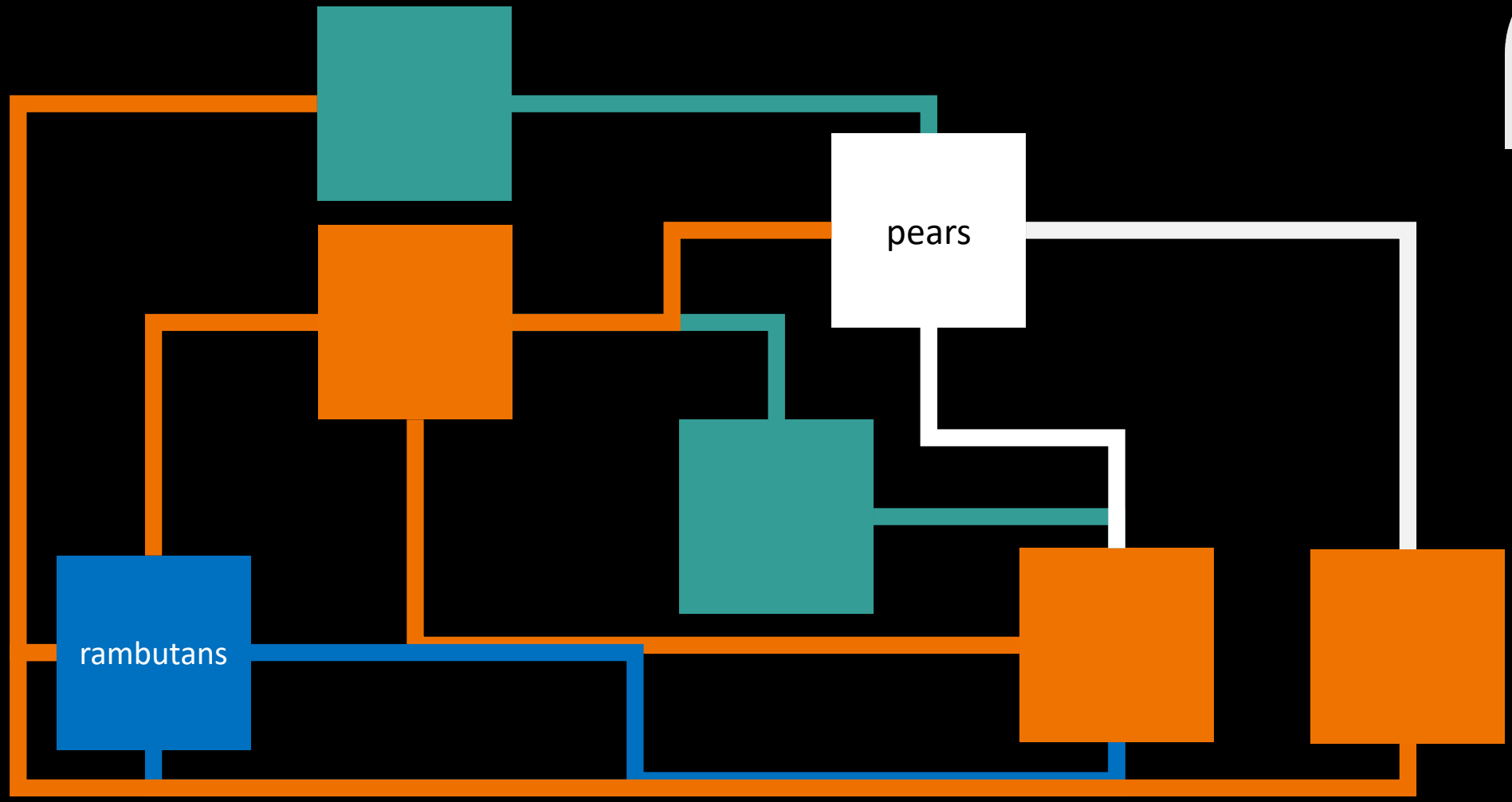




When we had just one type of fruit, integration was easy.

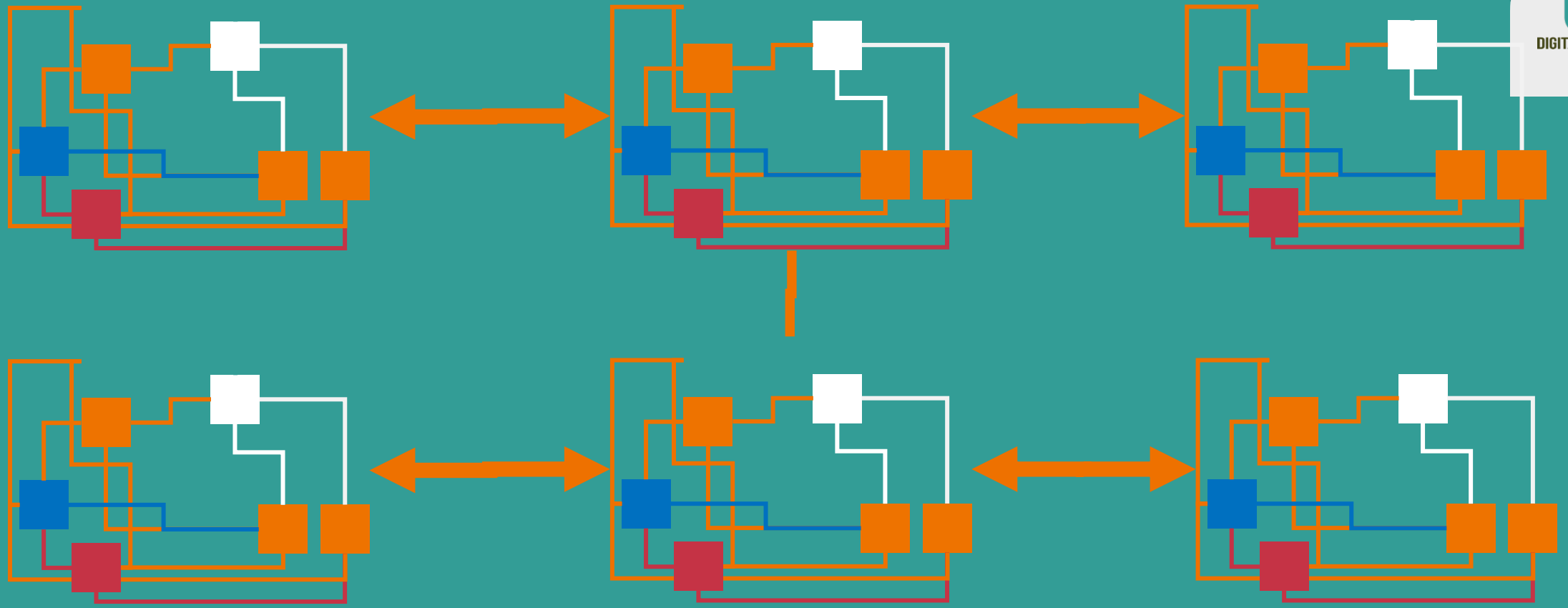


It got difficult when we started to add new fruits to the mix.



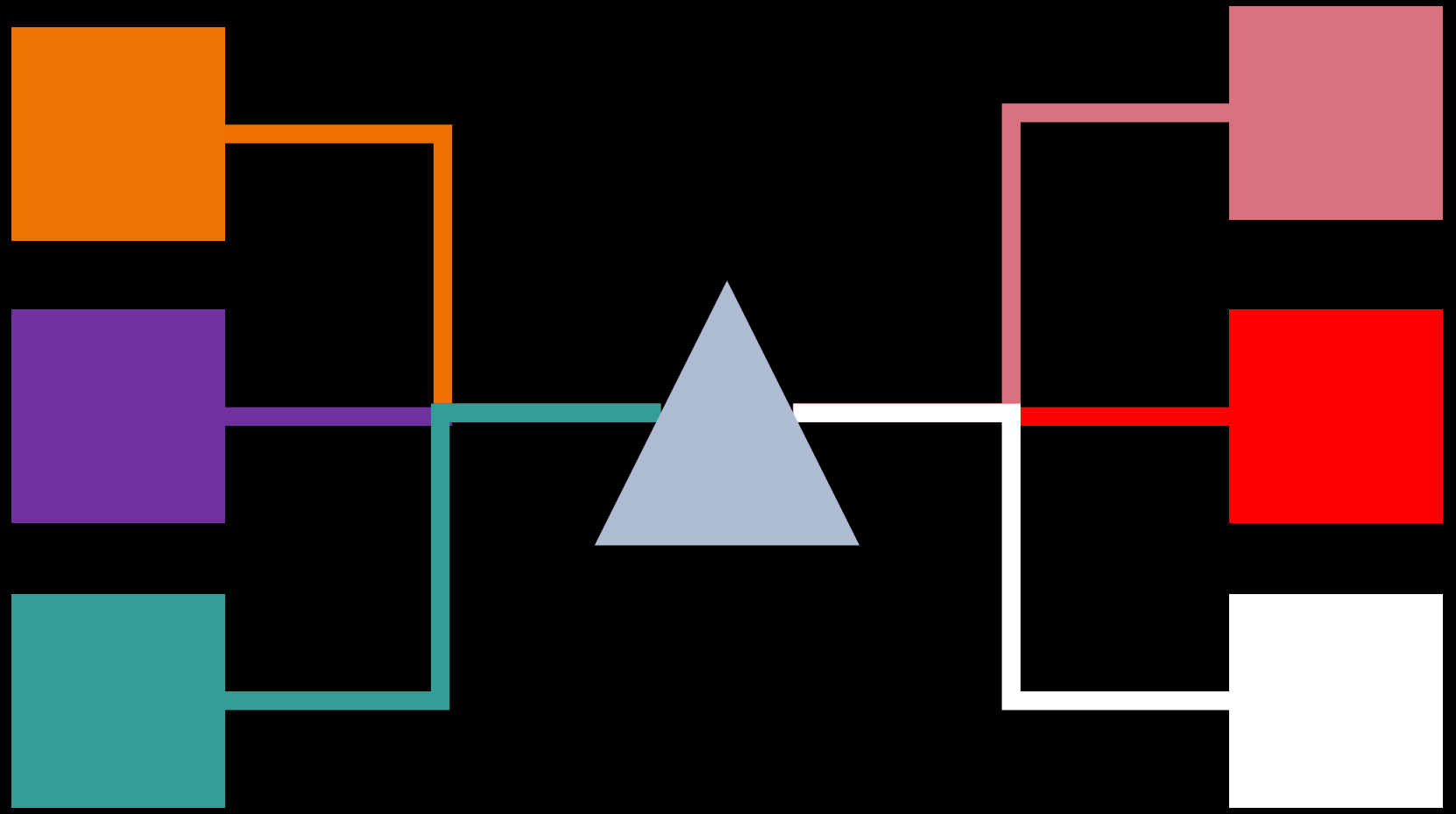
And then we added more new and also exotic fruits.
Things got complicated.

After a while, it got expensive and smelly.



The API-economy

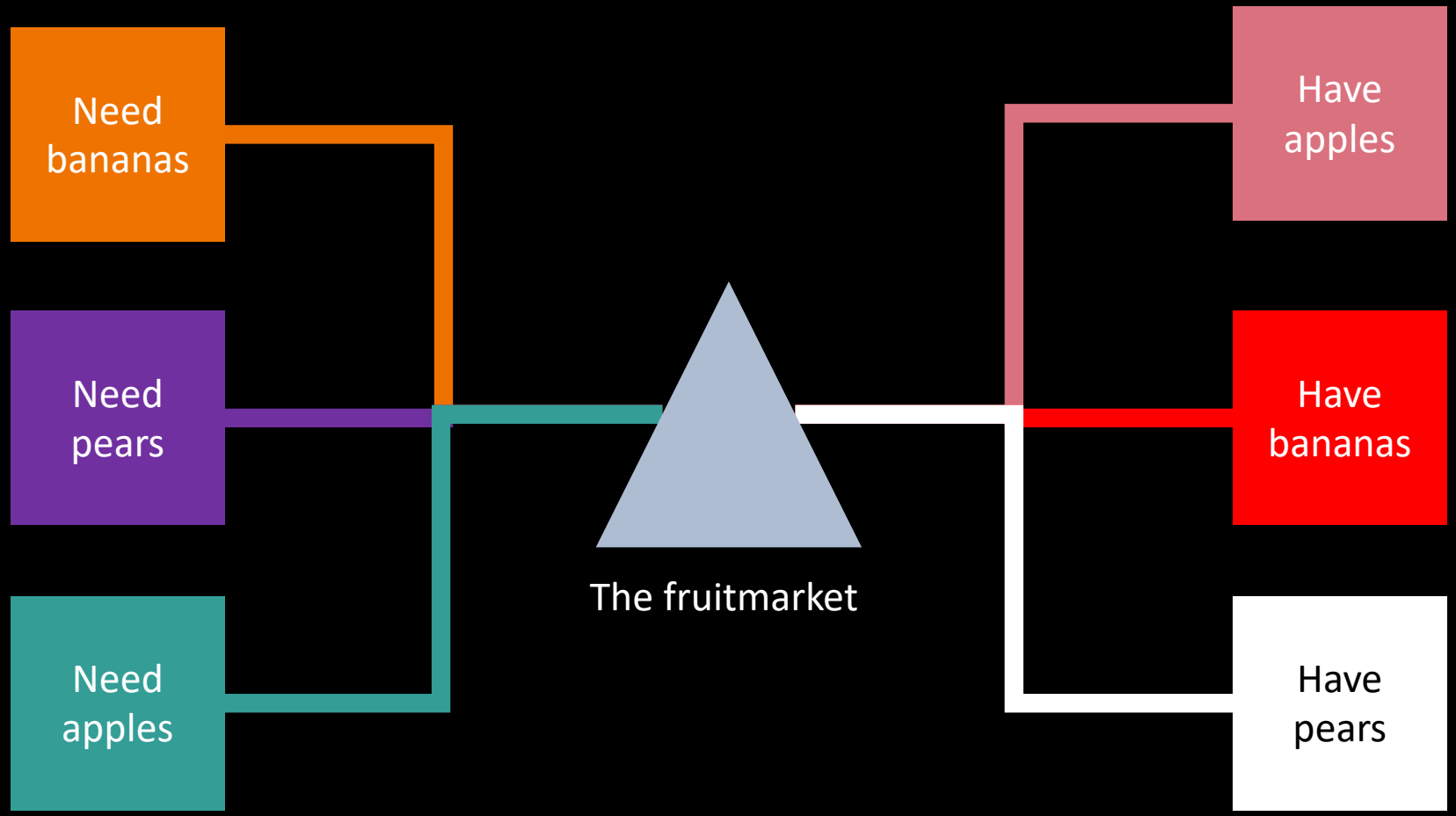
And in the meantime companies wanted to integrate all of eachother's fruits. The world started to look a bit like a fruit mayhem.



DIGITAL INNOVATION

Smart people found a solution.

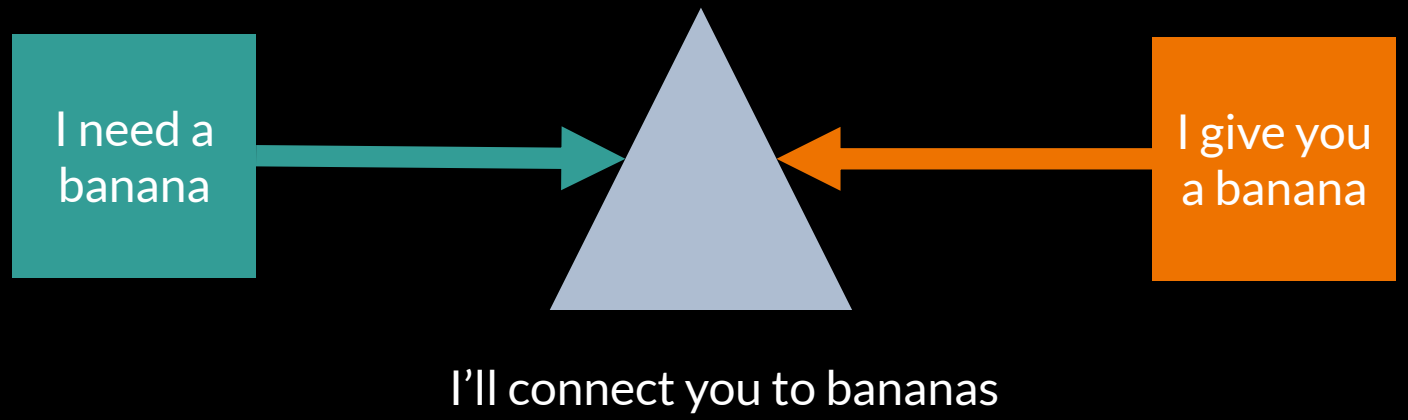
They invented a sweet, universal fruit paradigm that allows us to eat eachother’s fruits faster and better



Centralize services.

Use one central fruitmarket with all the fruit merchants instead of visiting all of the local fruit stands.

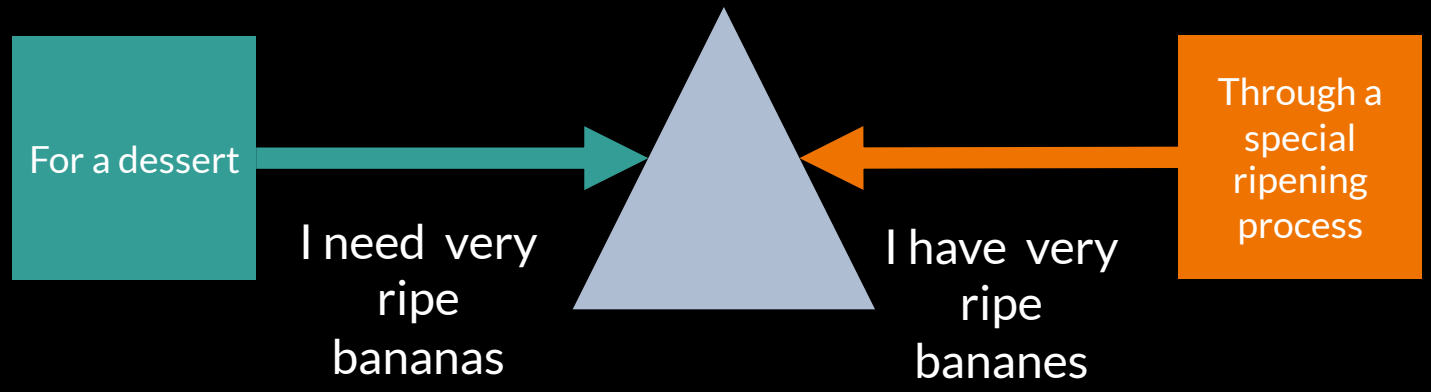
Instead of connecting everything-with-everything (also called point-to-point), we connect every-thing-to-one-thing and then make new connections with the one thing. This one thing is also sometimes called middleware, which can be an ESB.



Consumer-Broke-Provide is the key pattern for working with services.

The fruitmarket is where fruits are exchanged, consumed and supplied.

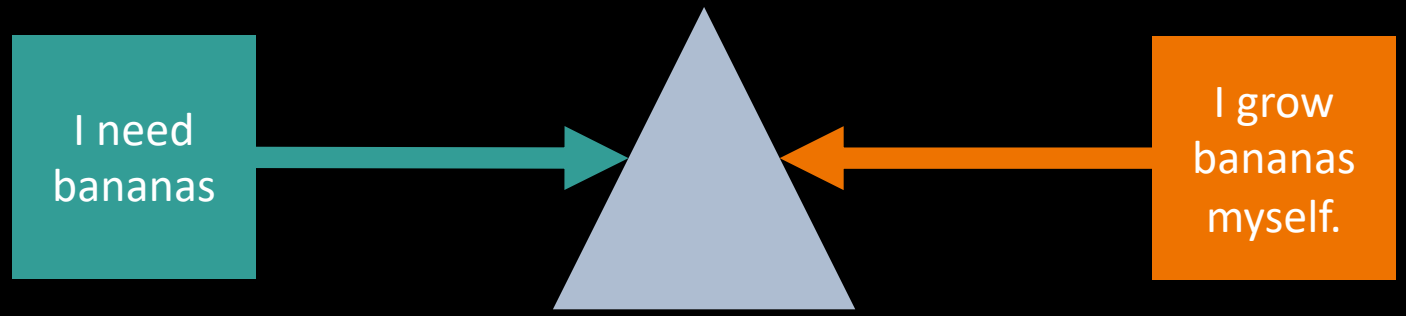
Think consume-broke-provide. In the middle there's a tool that will help you as a consuming application to find your data (through a service) in other, providing, systems.



Services should be hiding away internal logic – the systems behind services should be blackboxes.

You don’t need to know how to grow fruit in order to consume it.

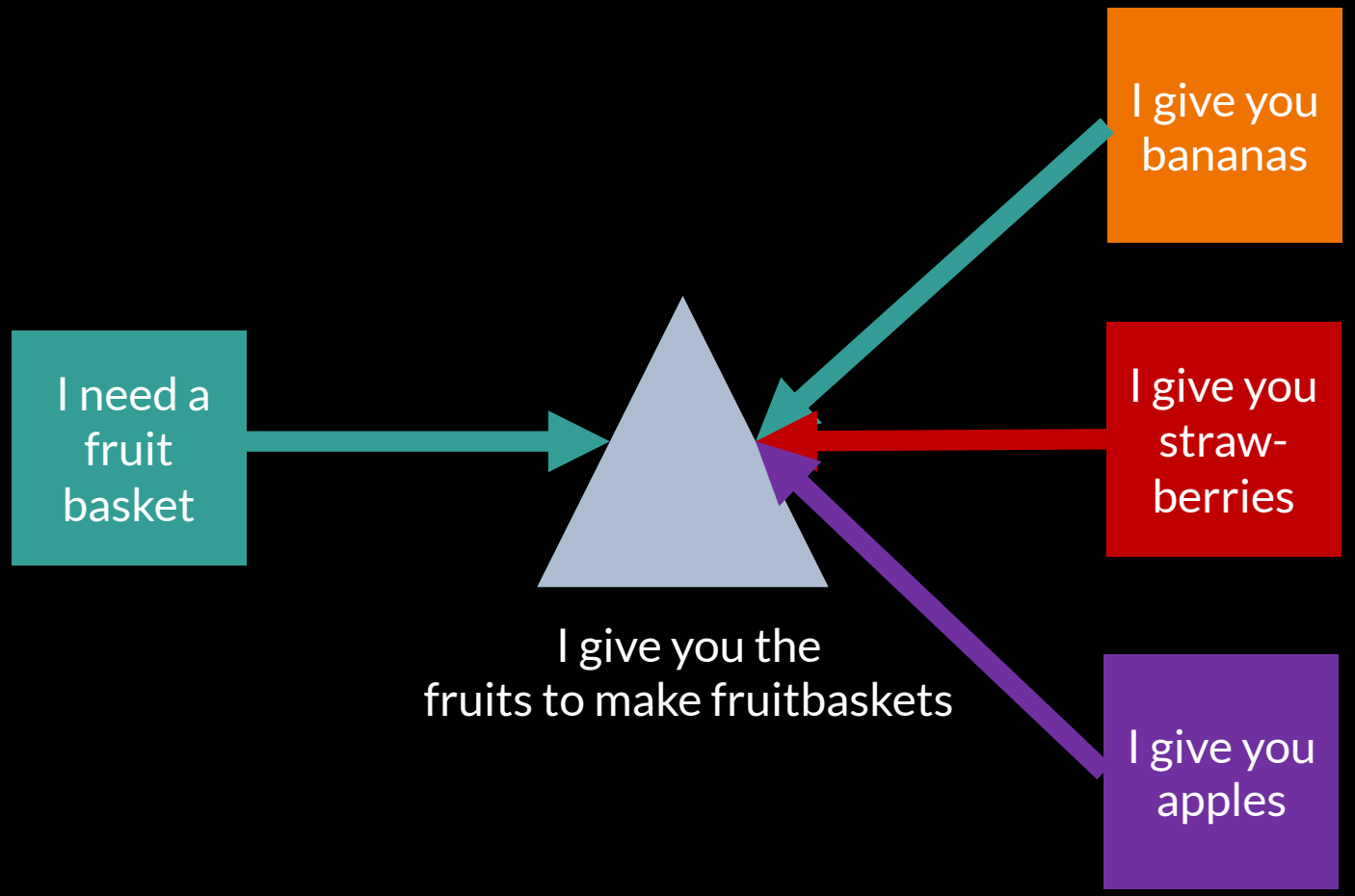
Avoid the need to understand what’s inside the other system in order to build stable and sustainable integrations with that system, by standardizing interfaces.



Services should be autonomous.

If you provide fruits, make sure you can grow them yourself.

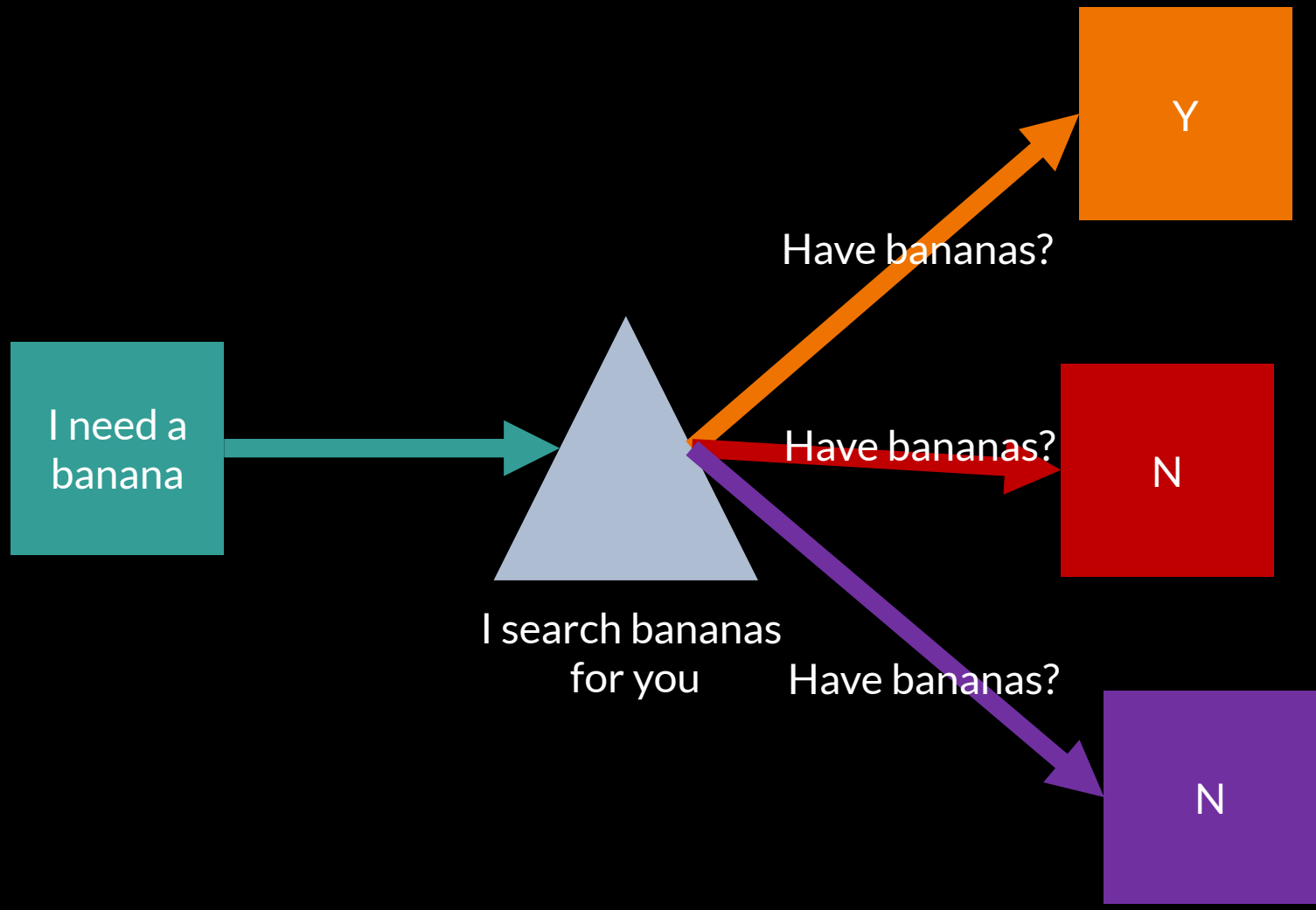
Make sure that the information that you provide as a service is not depending on other systems outside of your system itself.



Services should be composable.

If you want to create fruitbaskets, search and combine the different fruits–don't start growing the fruits yourself.

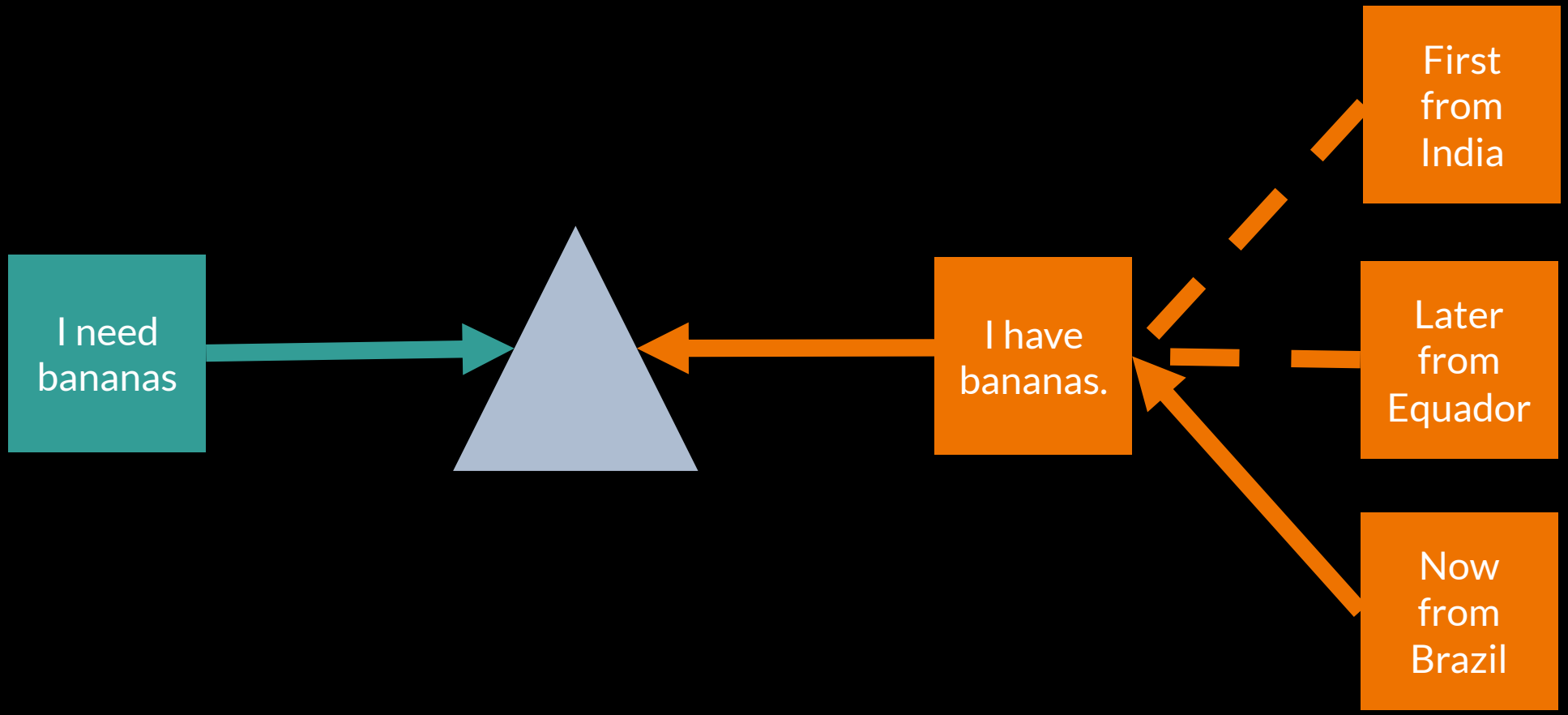
Services need to be composable so that smaller services can be combined into larger services that provide a specific value.



Services should be discoverable.

Label your fruits so people can find them and understand what they are.

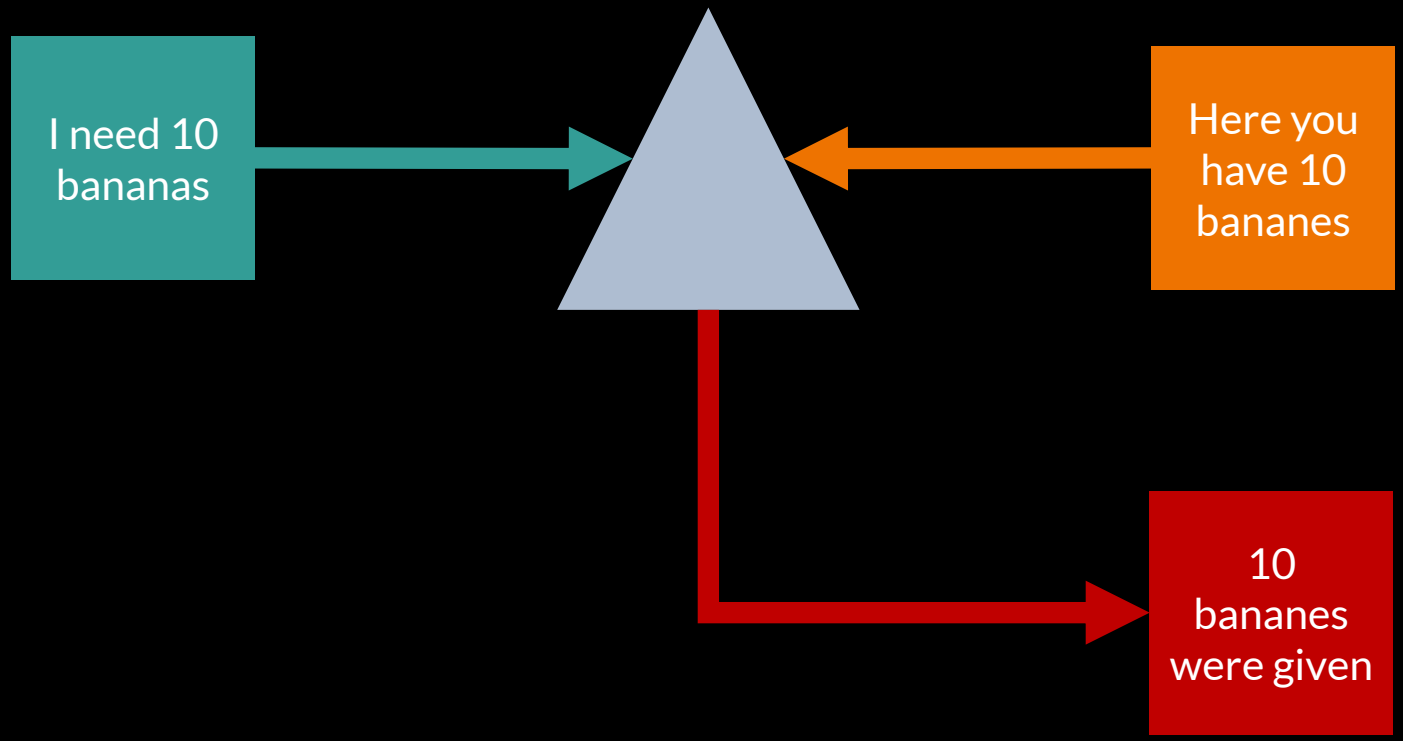
Make sure that services can be discovered and that they are meaningful in their function. This way, re-usability is stimulated and redundancy is lowered.



Services should be build for longevity.

If customers know you because of your bananas, make sure you can continue to offer bananas.

Services should be designed to be long lived. If you call a service today you should be able to call the same service tomorrow.



Services should be stateless.

When you sell fruit, keep your stall clean and about fruit – not about your fruit business.

Services should be stateless, that is never storing any state information or transaction through the service itself.

Get in touch with us.

Digital Innovation
antwerp@digital-innovation.com