

DDD & Microservices

At last, some boundaries!

Eric Evans
@ericevans0
domainlanguage.com

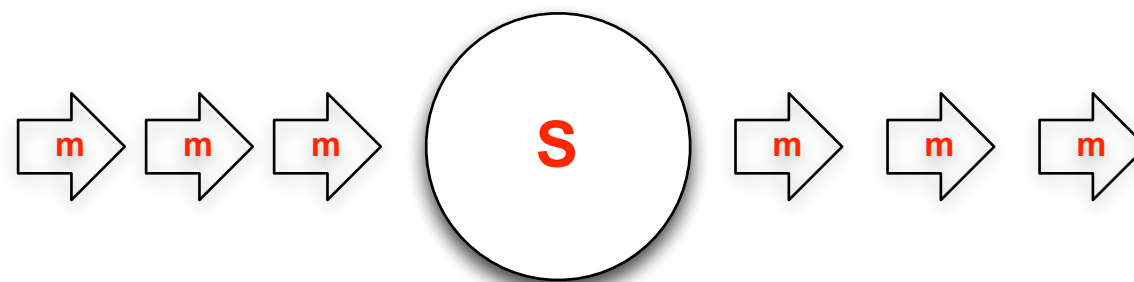
Why do I like microservices?

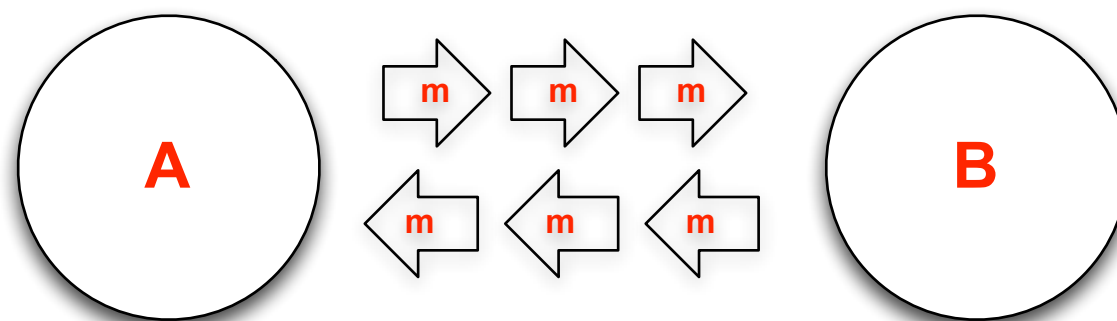
- Autonomous teams with isolated implementation.
- Acknowledge the rough and tumble of enterprises.
- Cattle not pets.
- A philosophical break from the past — gives us a chance to shake assumptions.

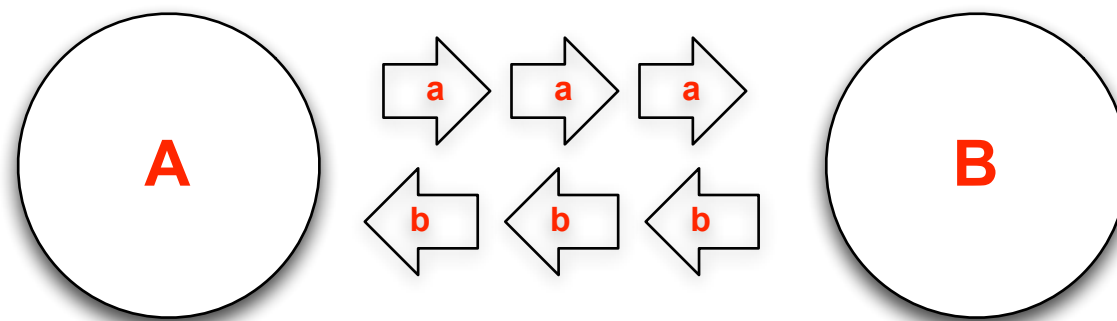
Possibly, some other people mean
something else:

What I'm saying may not apply to that!

Services and Messages



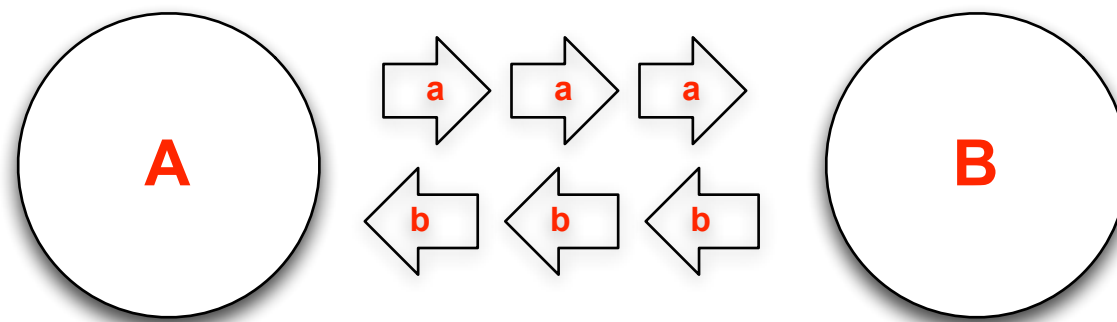




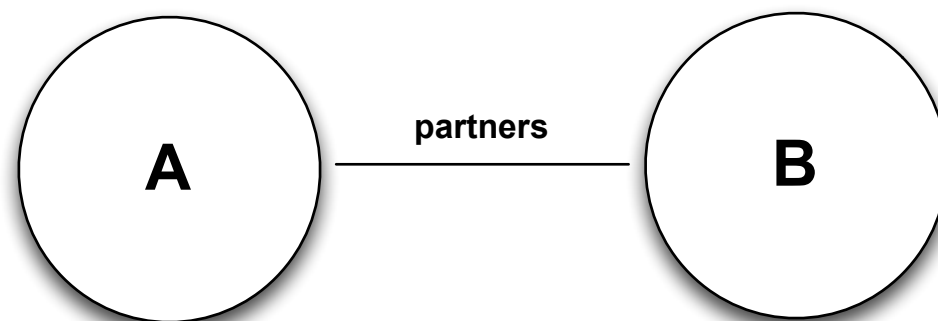
How do they understand the messages?

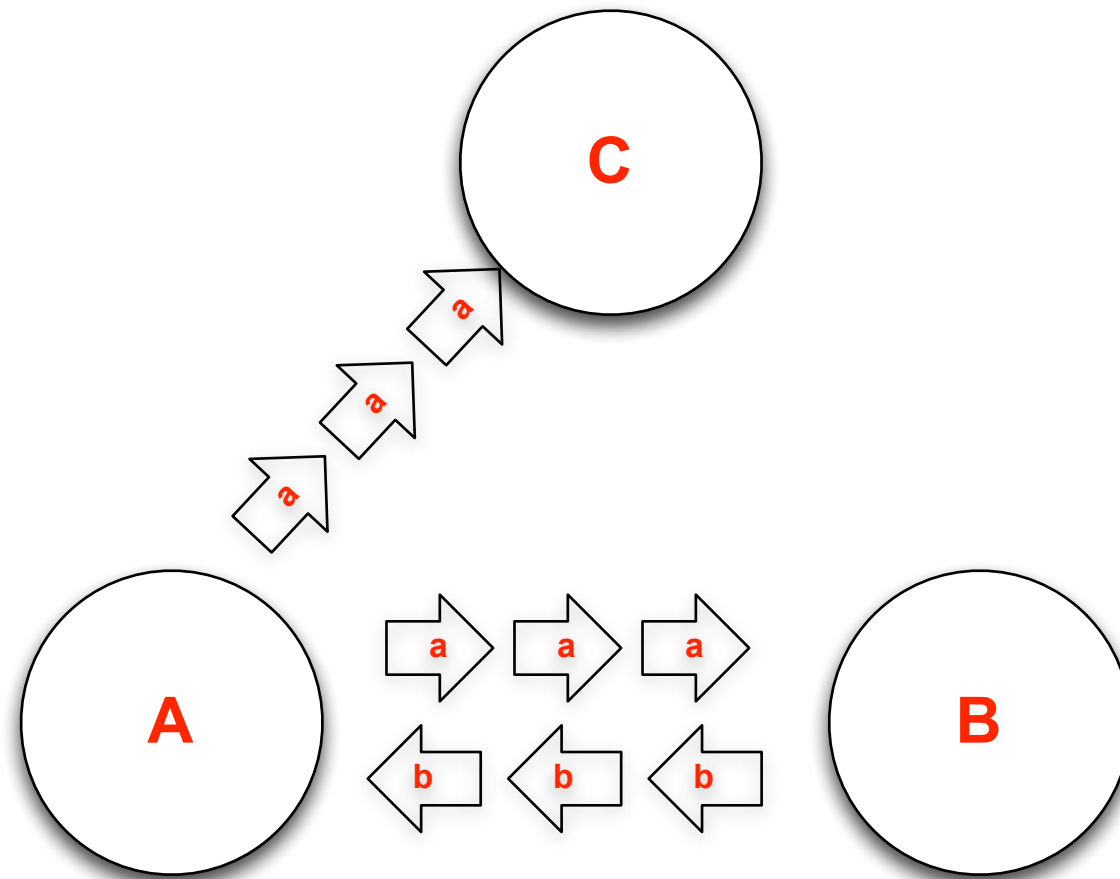
Bounded Context

- context The setting in which a word or statement appears that determines its meaning
- bounded context The conditions under which a particular model is defined and applicable.

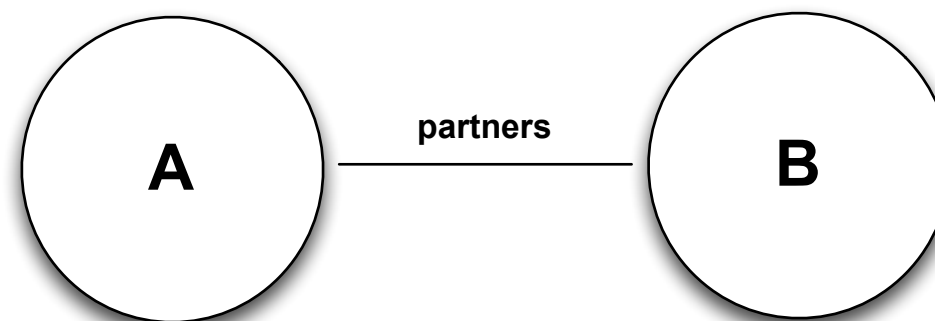


Context Map

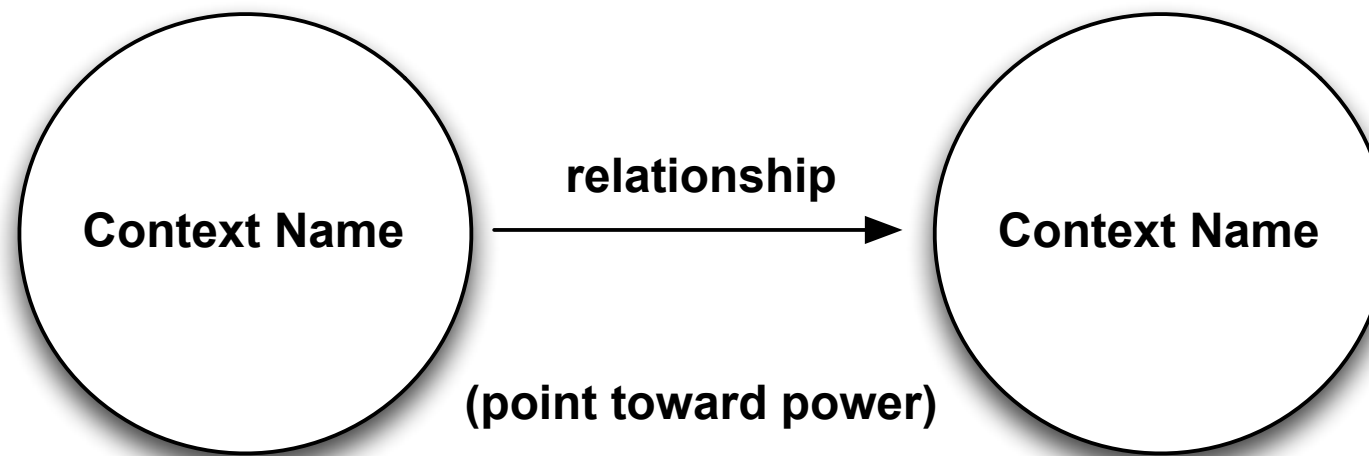


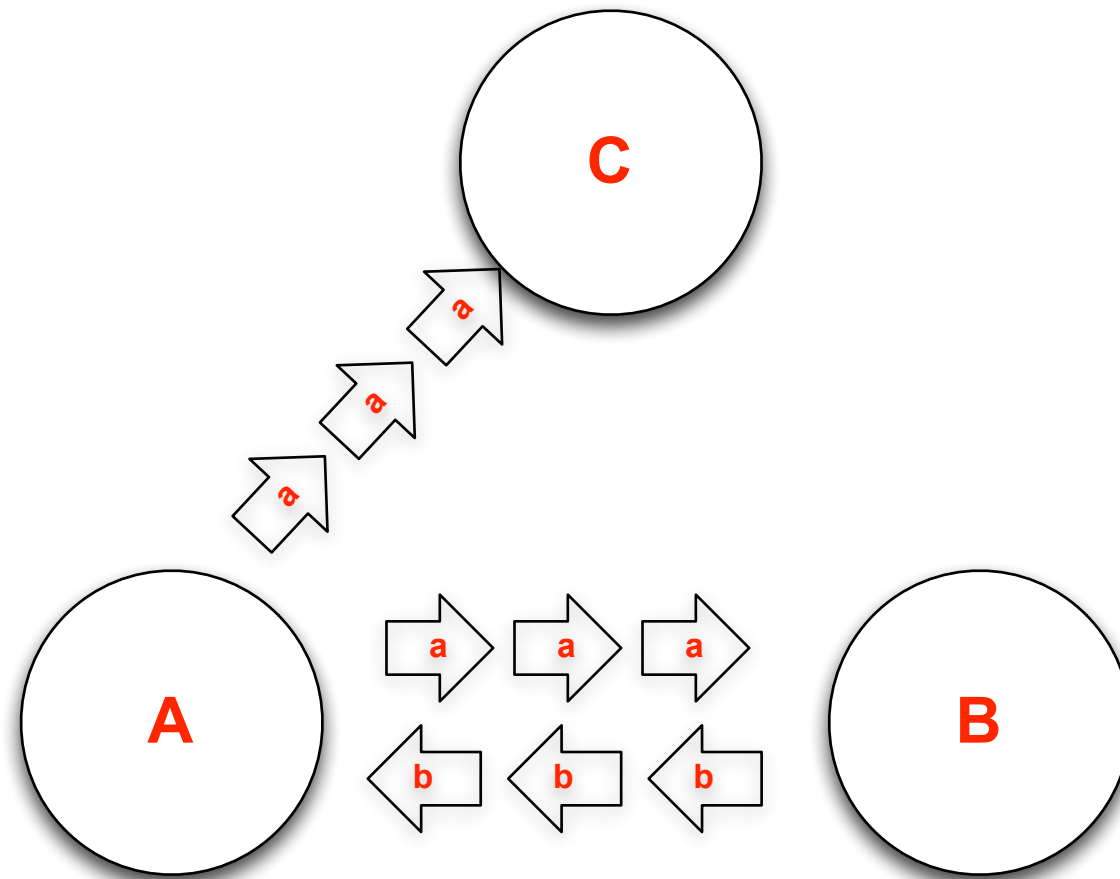


Context Map

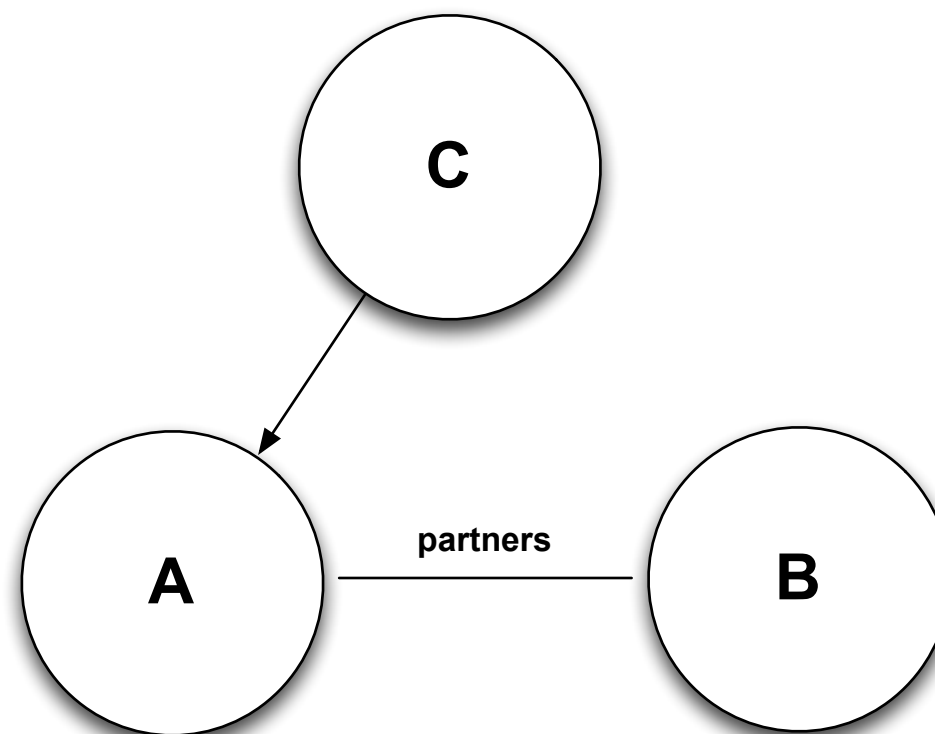


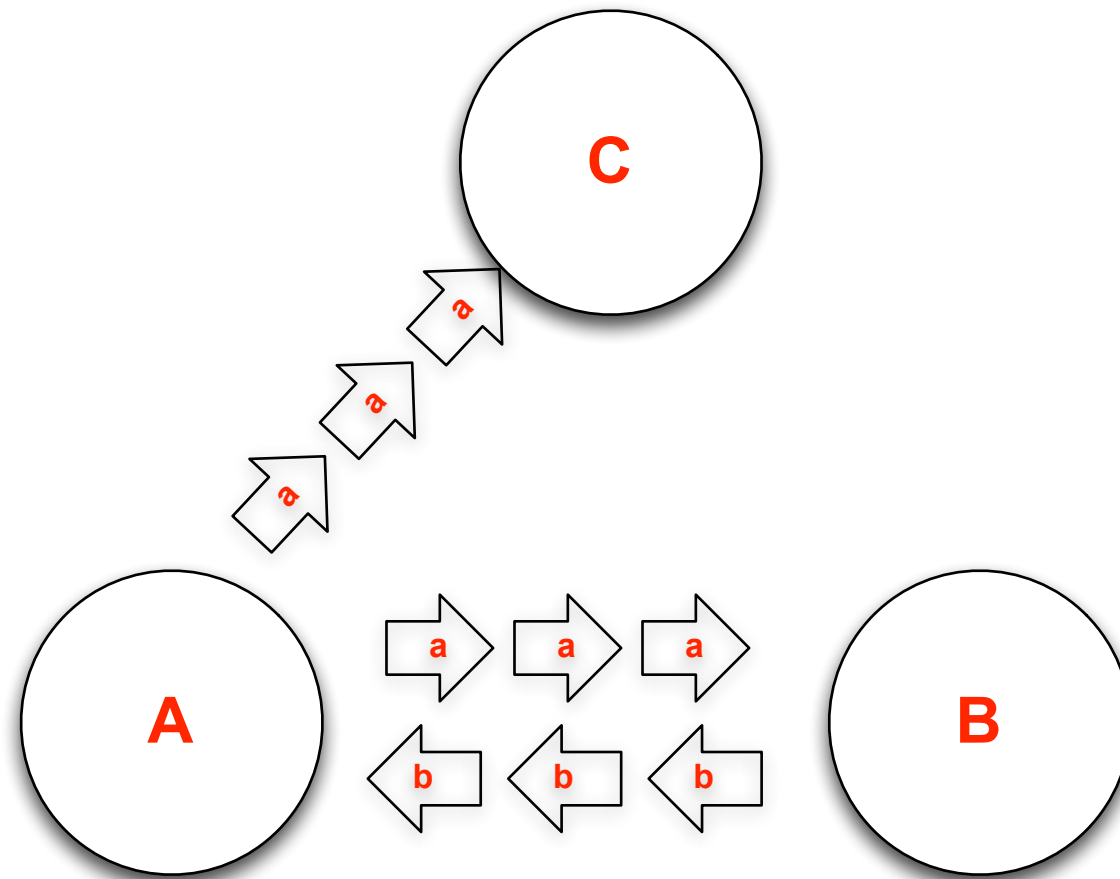
Asymmetrical Relationships



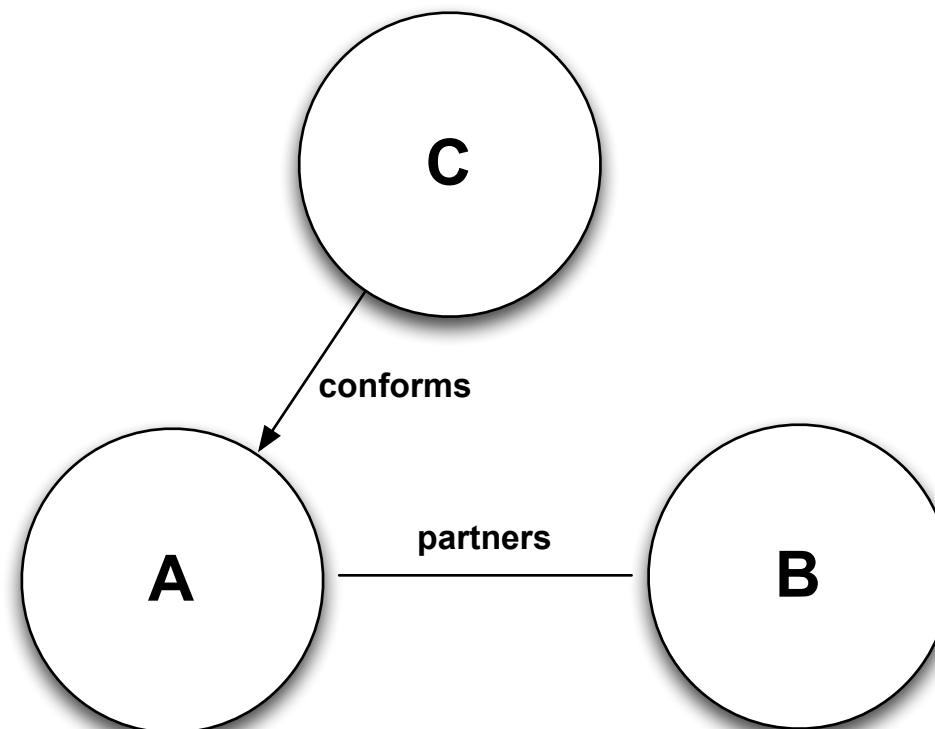


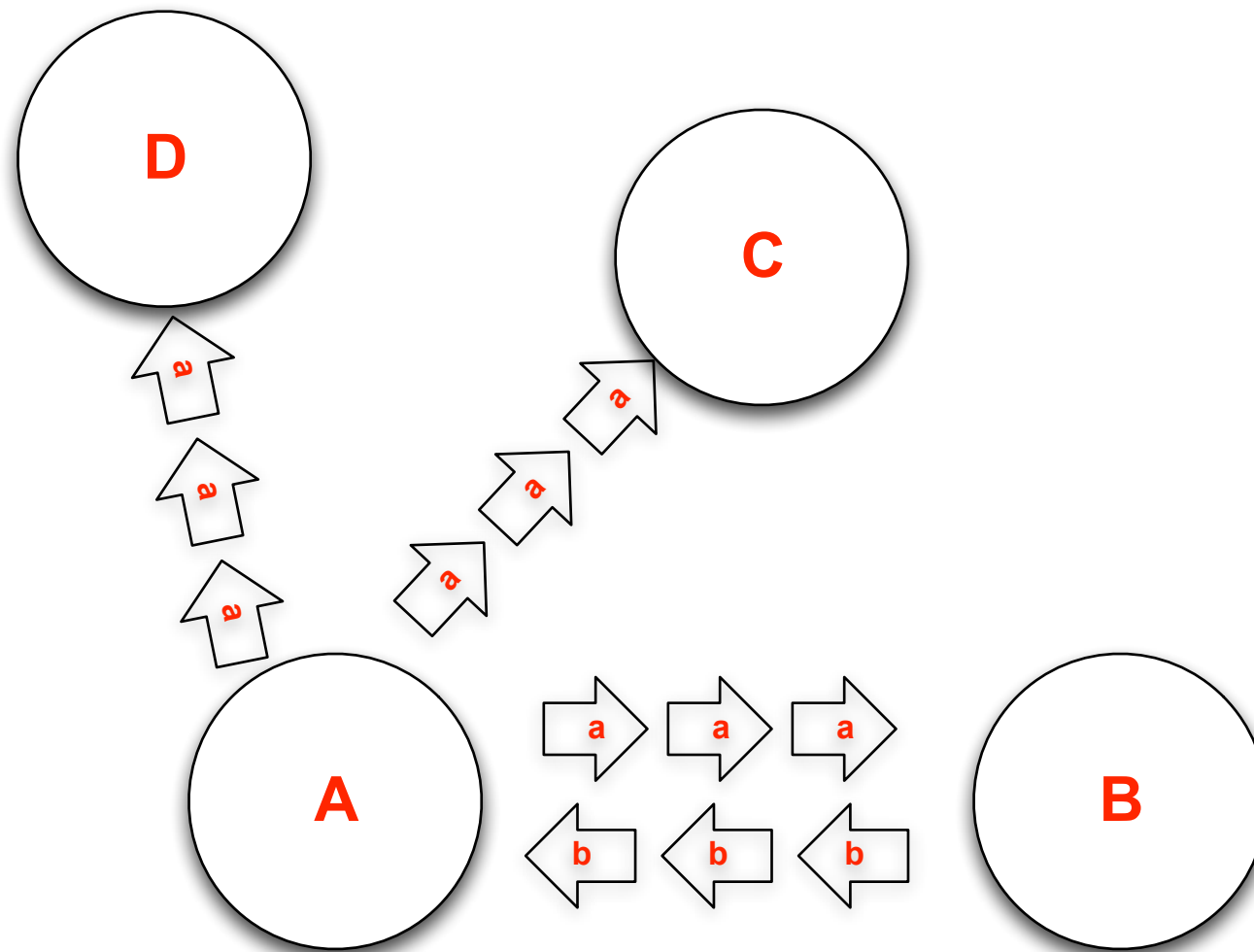
Context Map



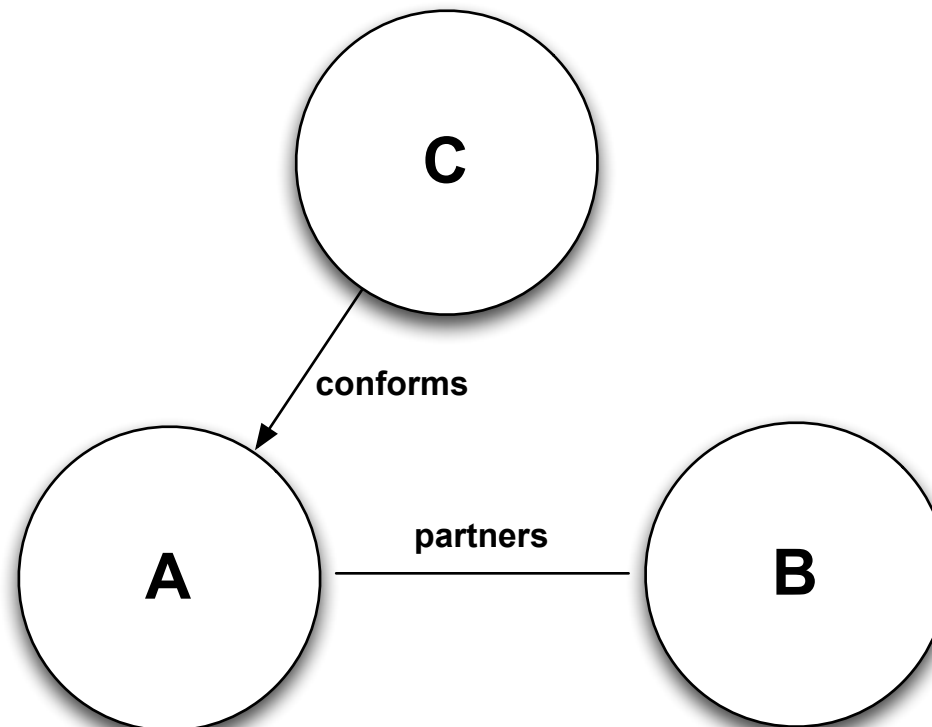


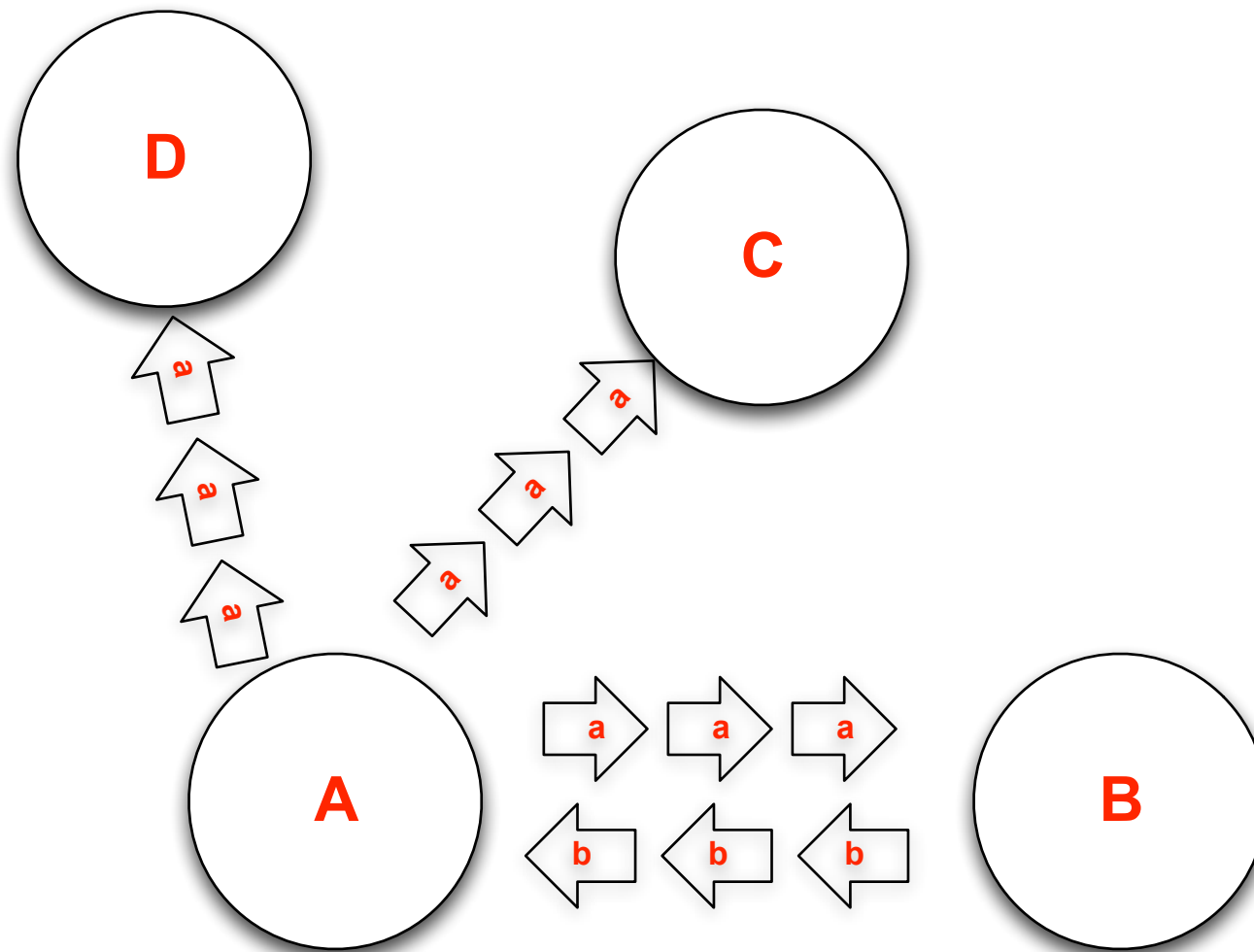
Context Map



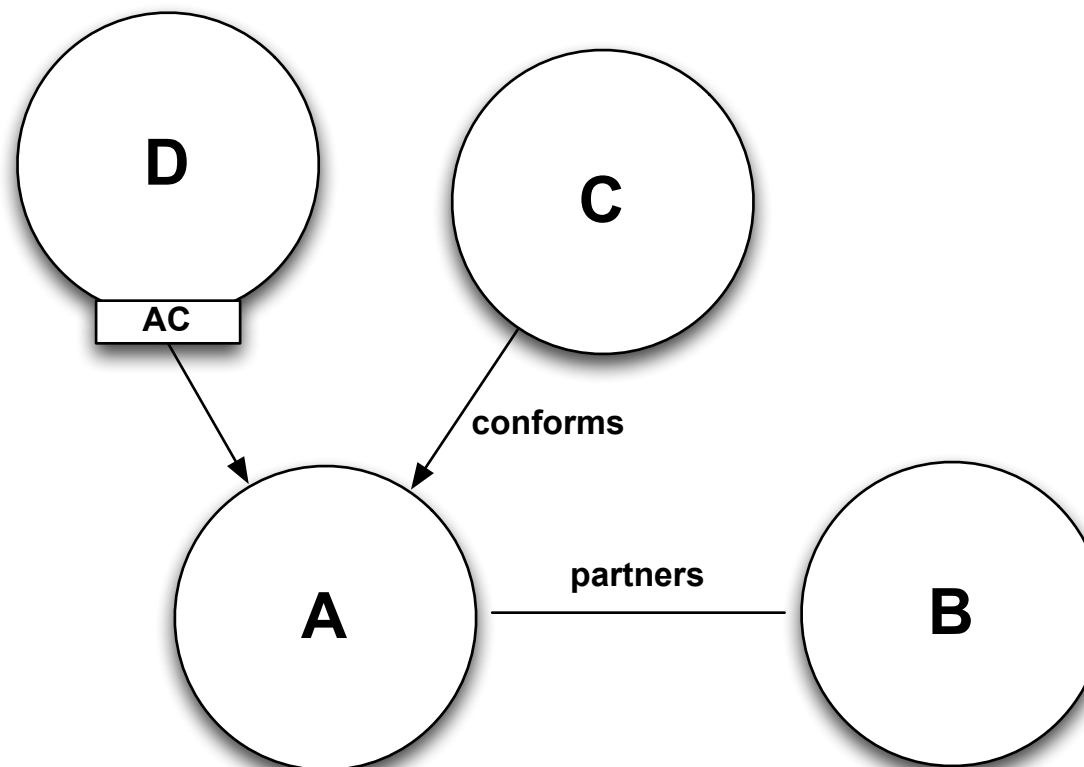


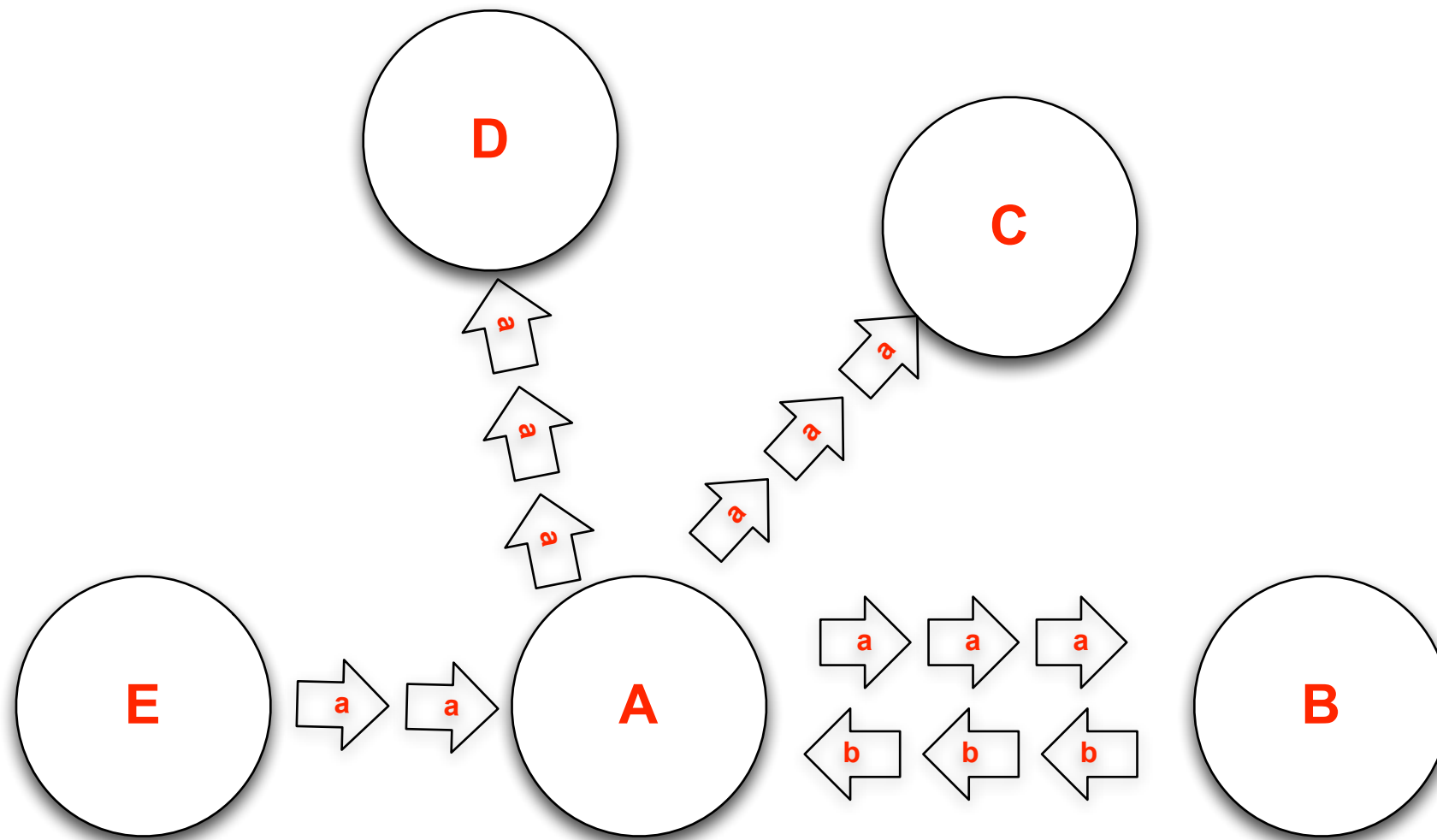
Context Map



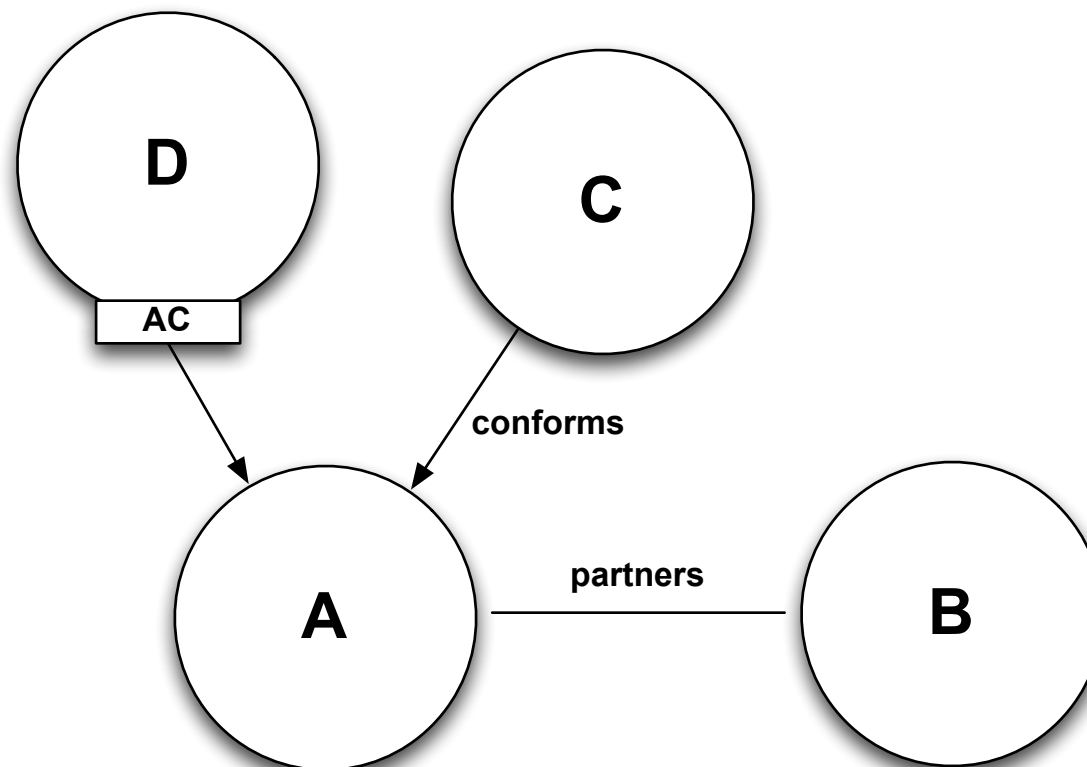


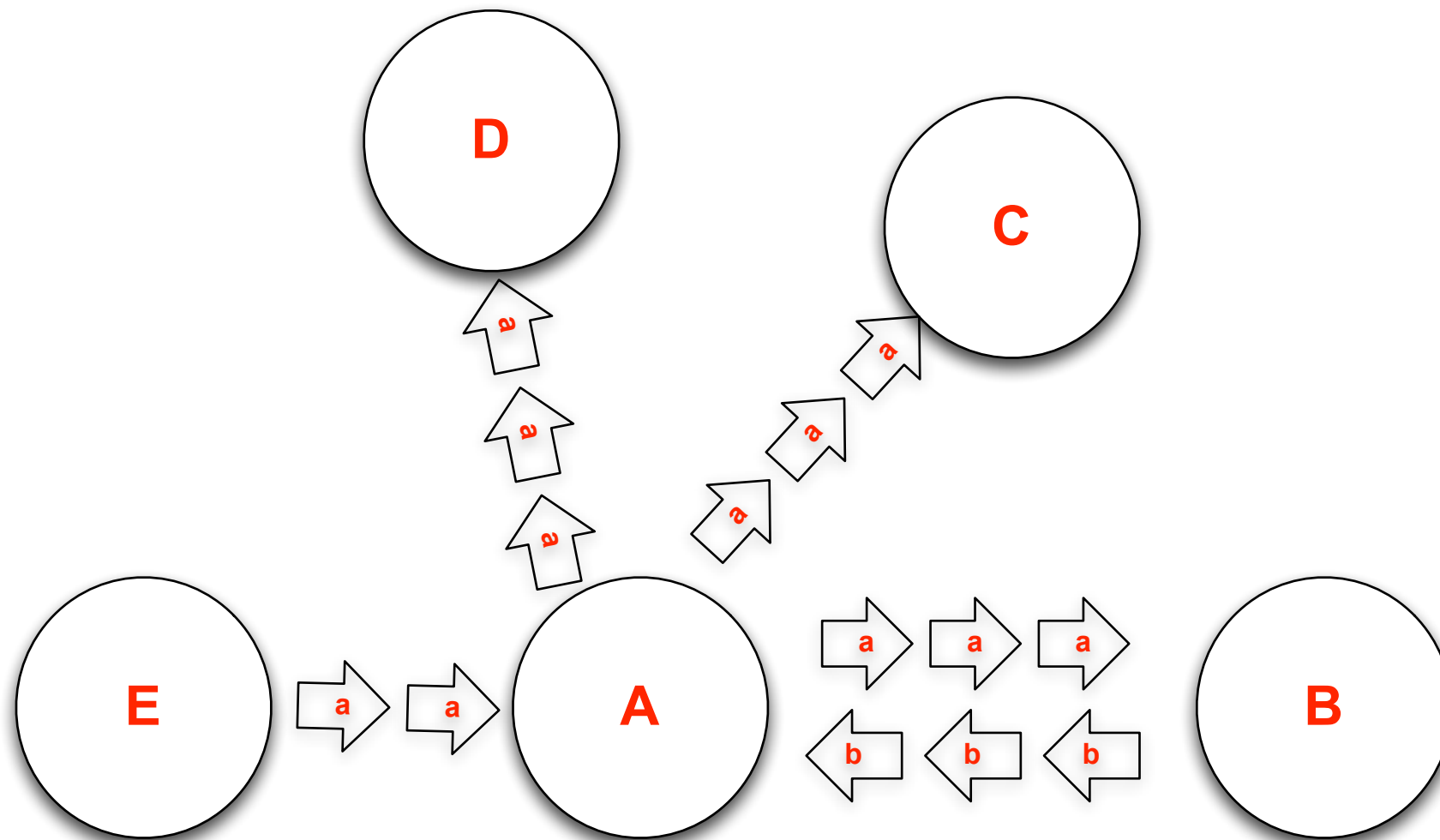
Context Map



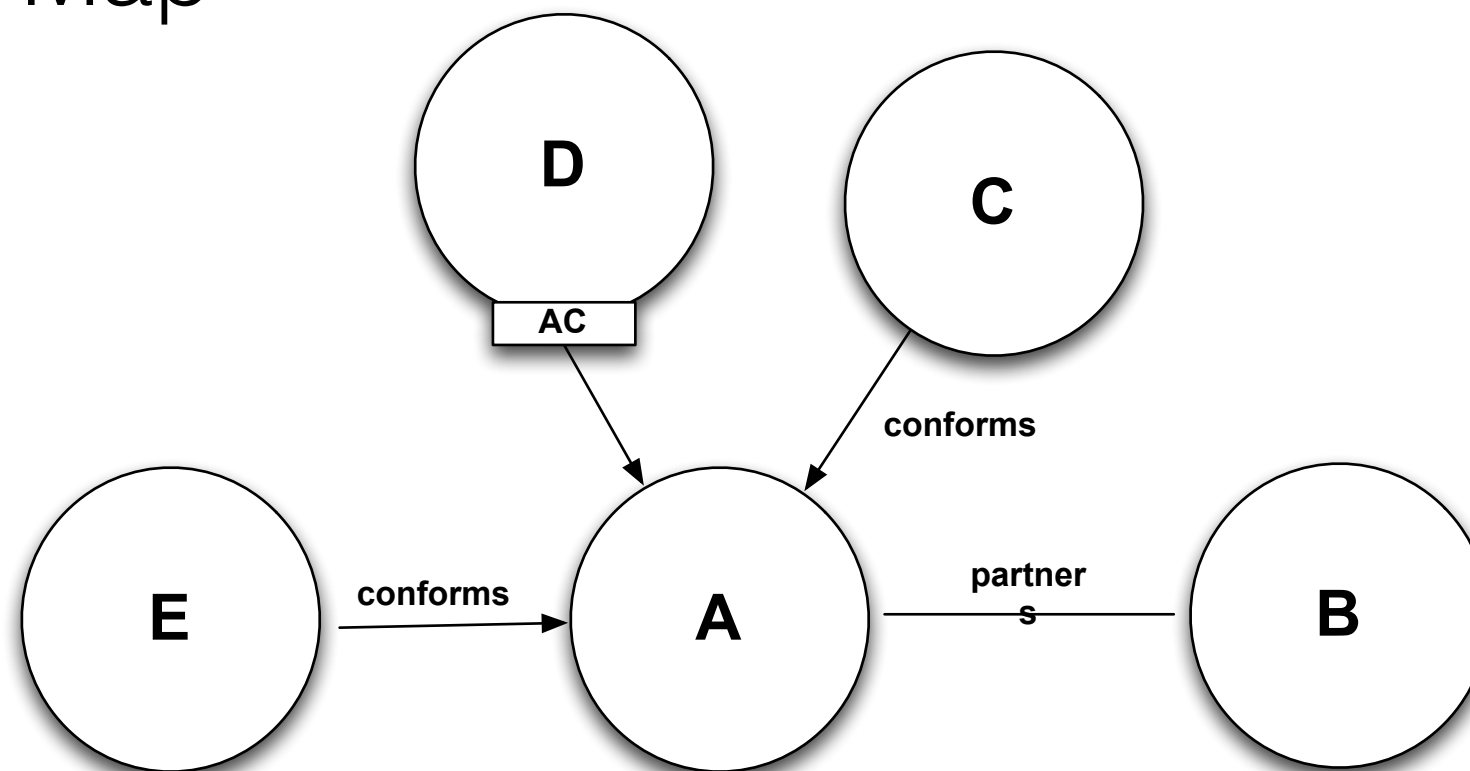


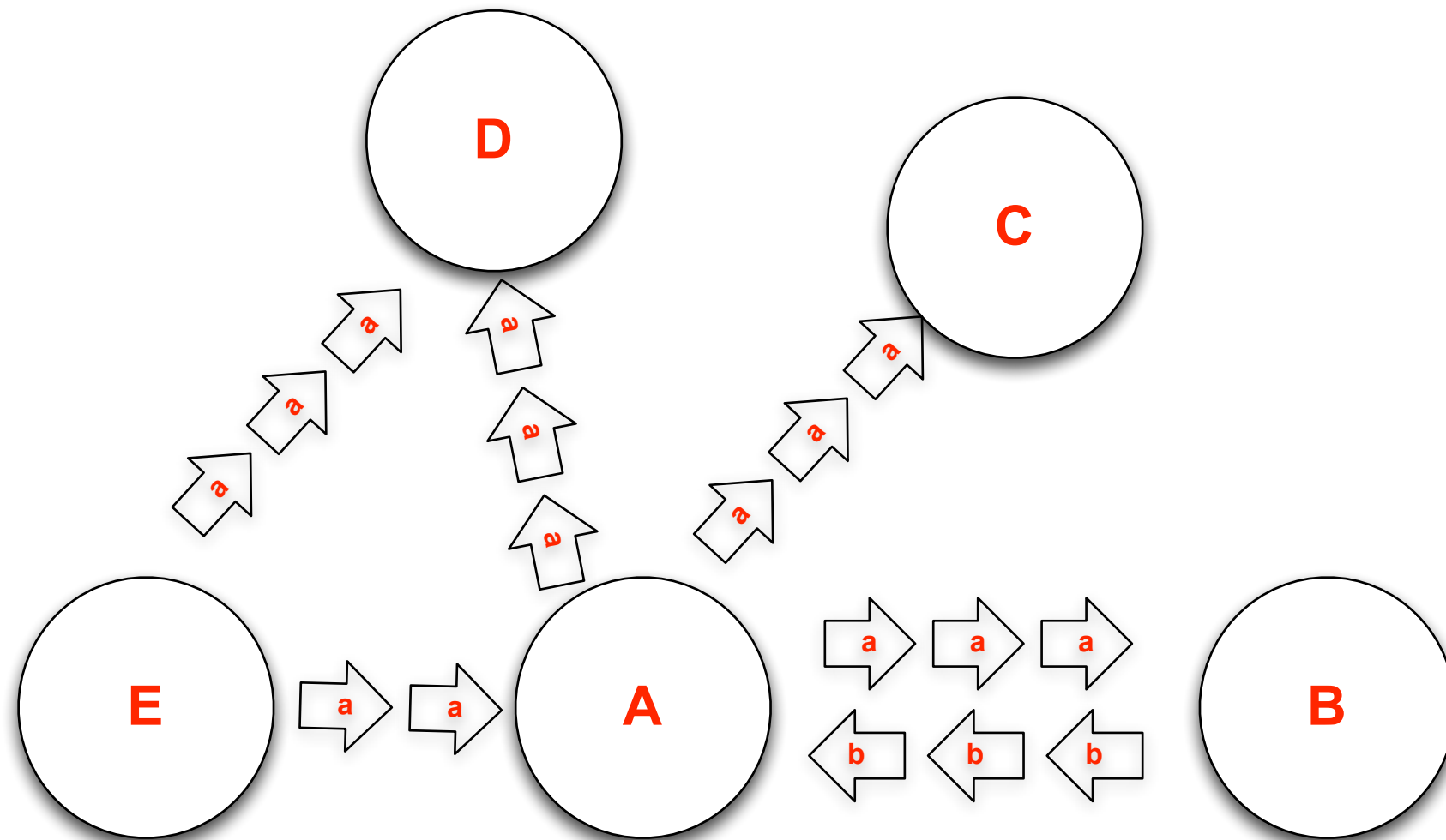
Context Map



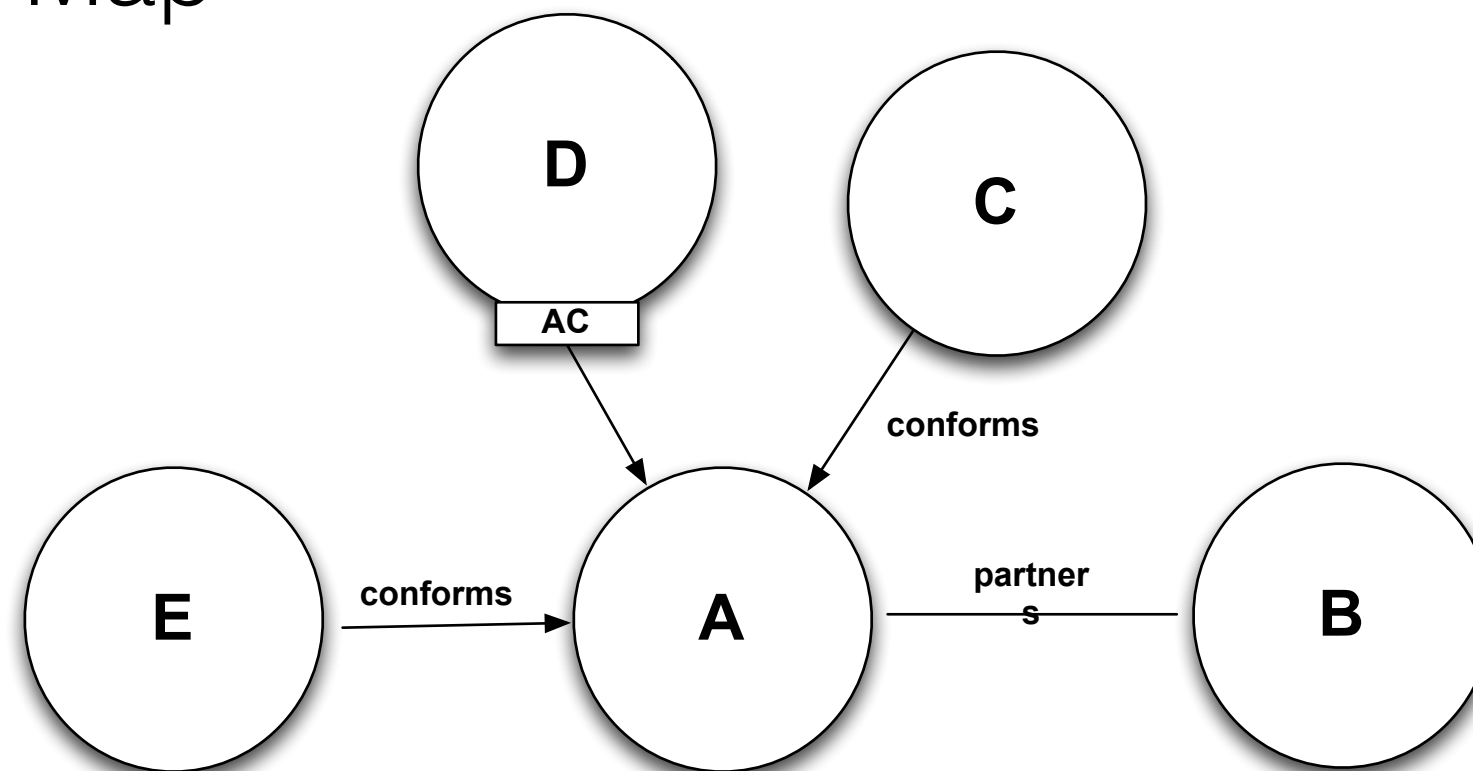


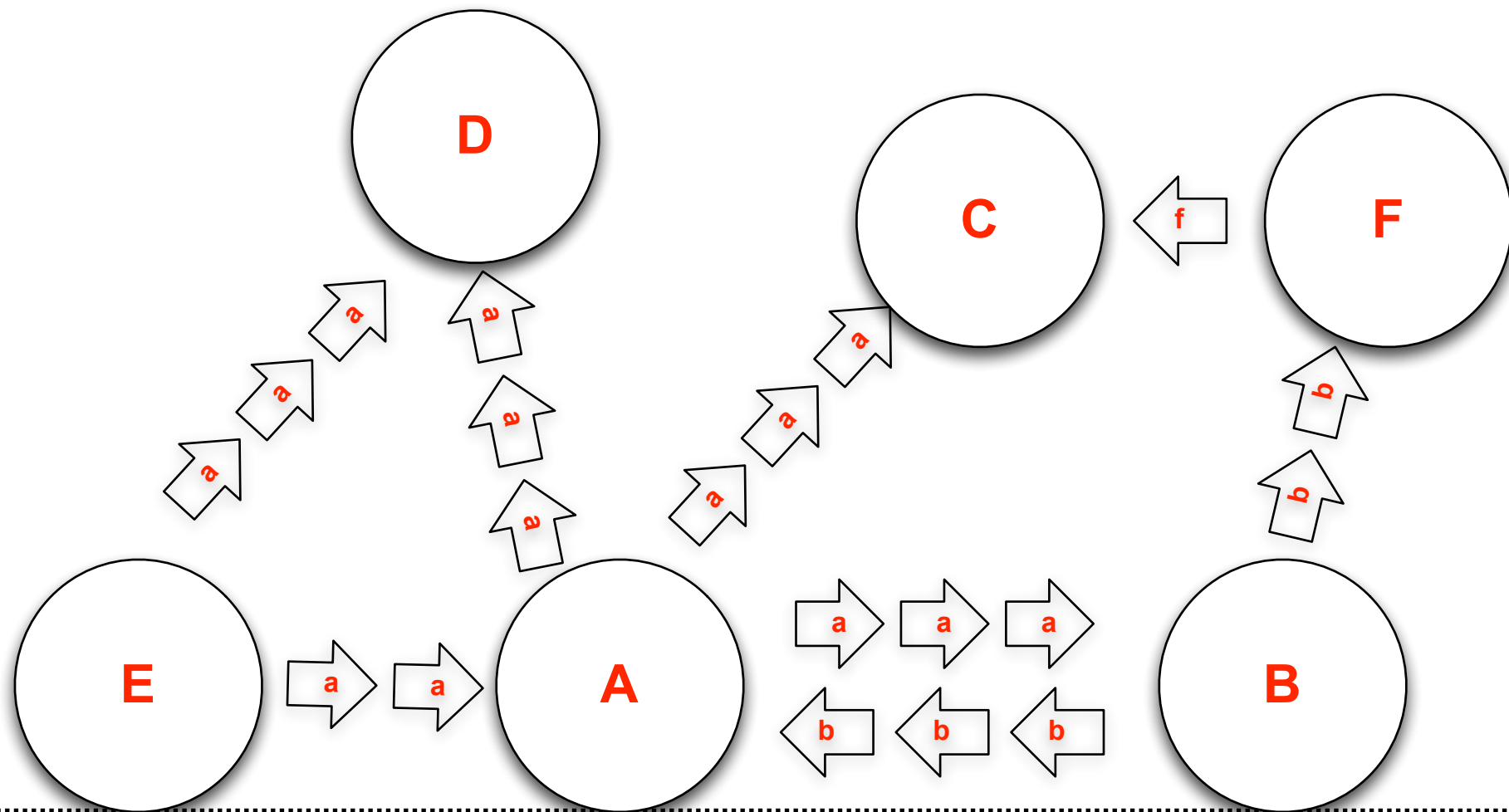
Context Map



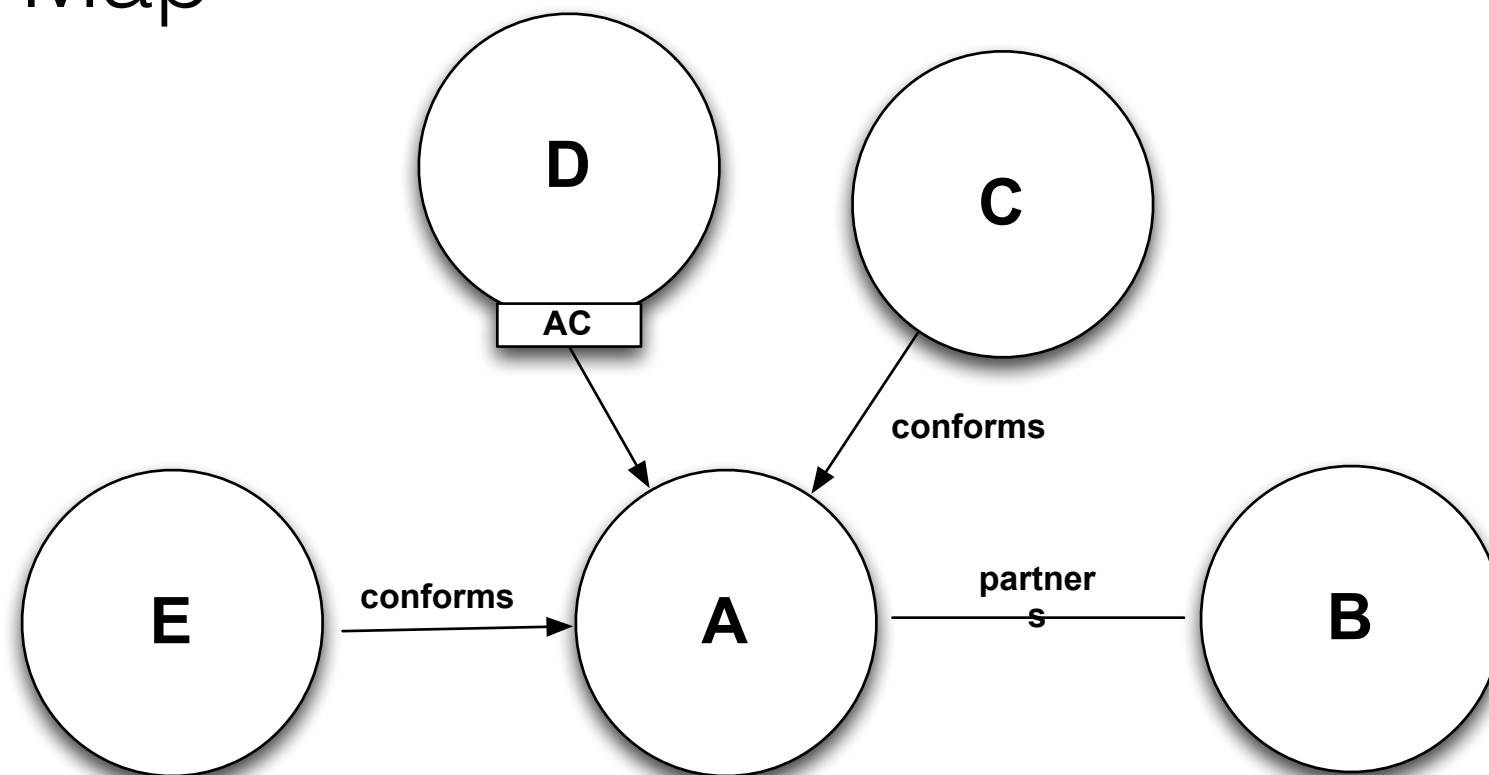


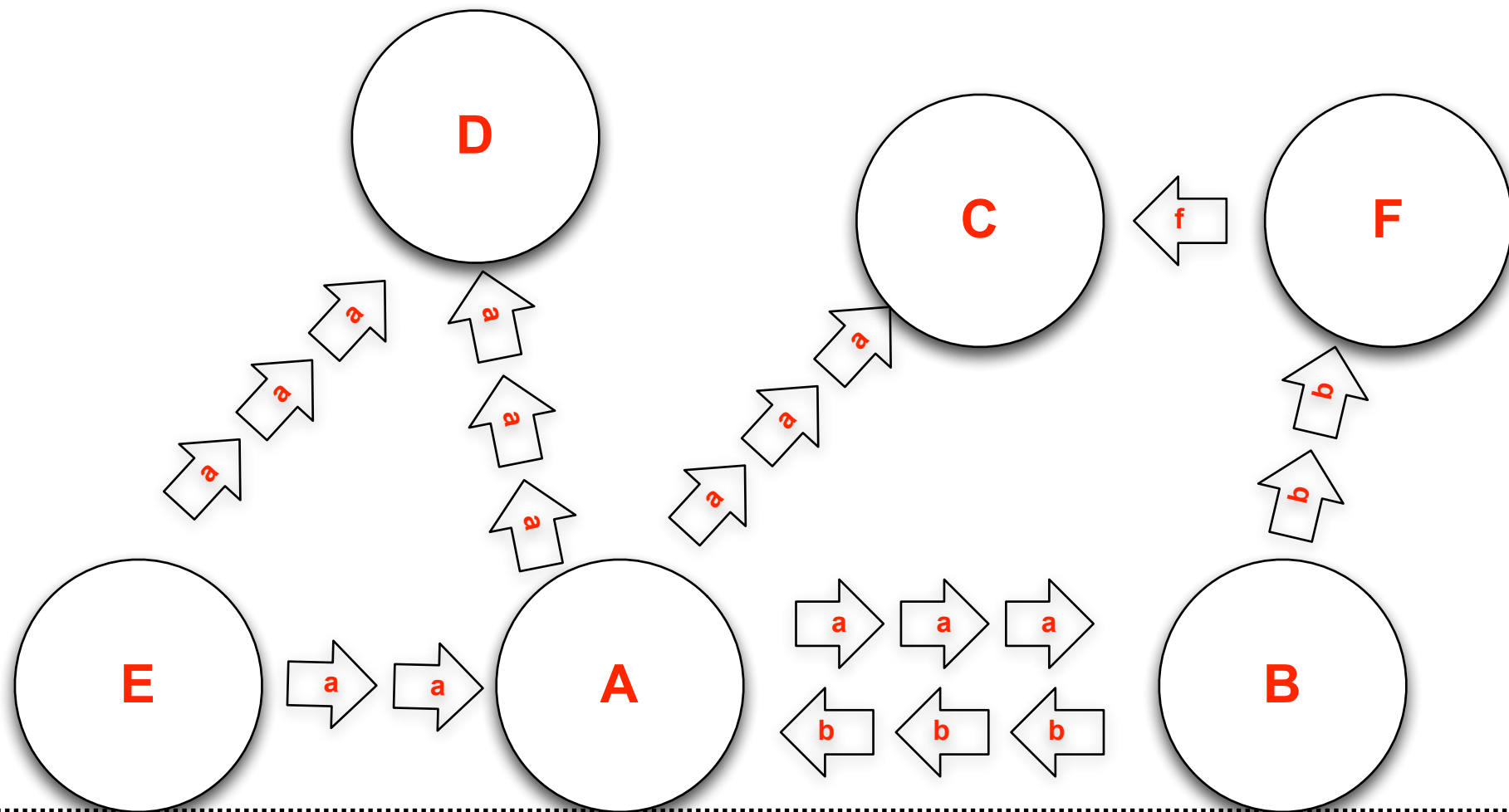
Context Map



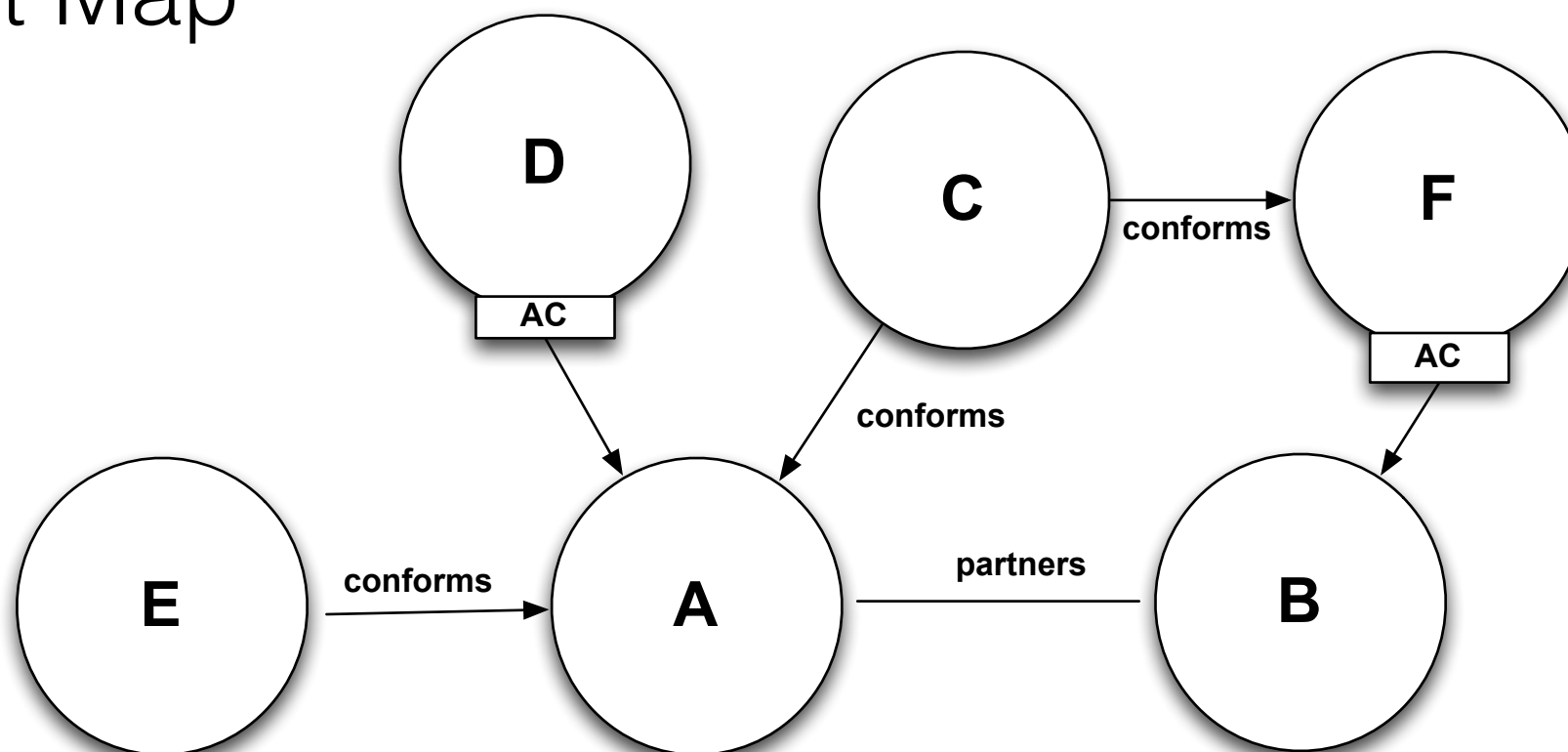


Context Map





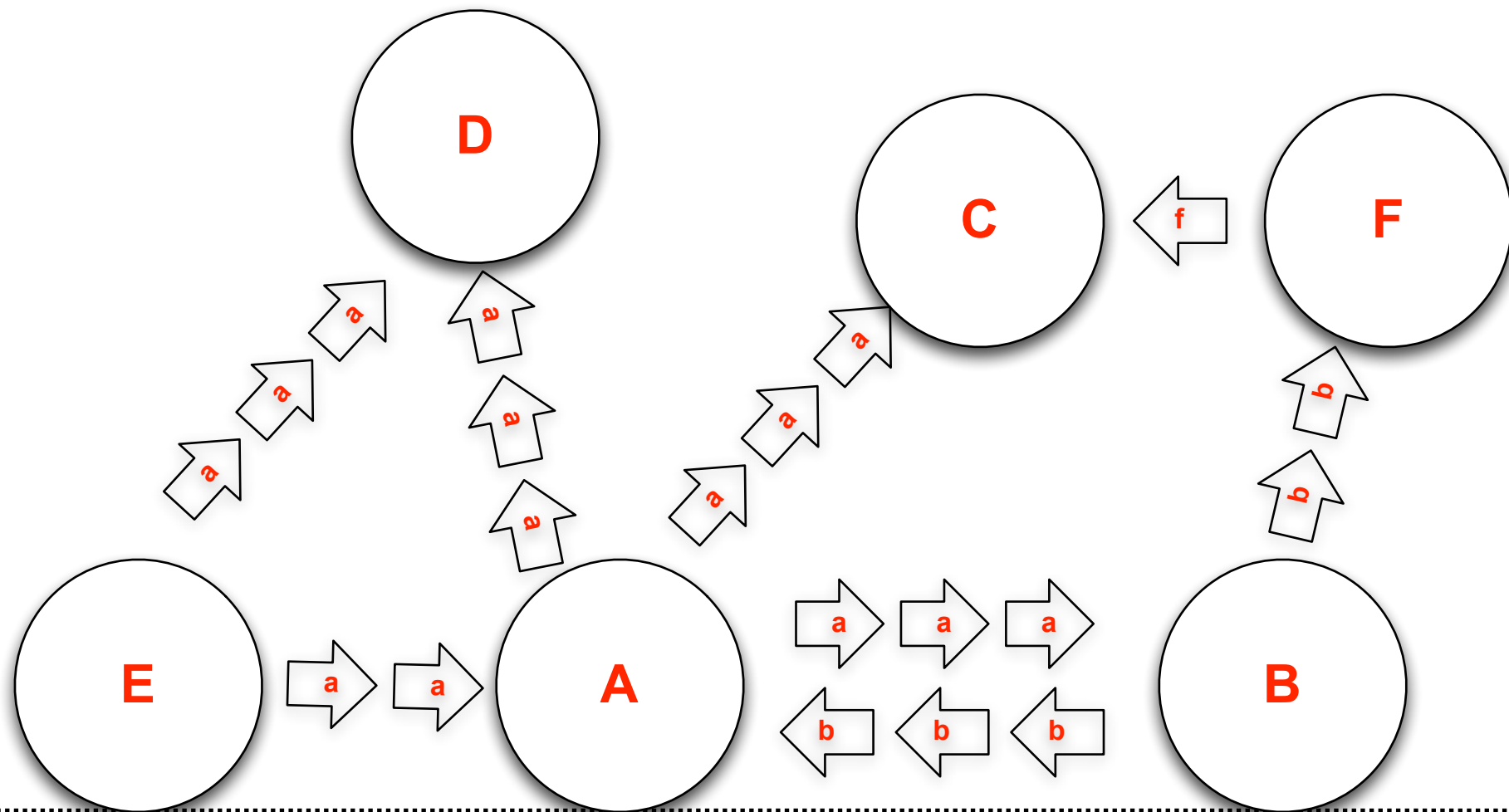
Context Map



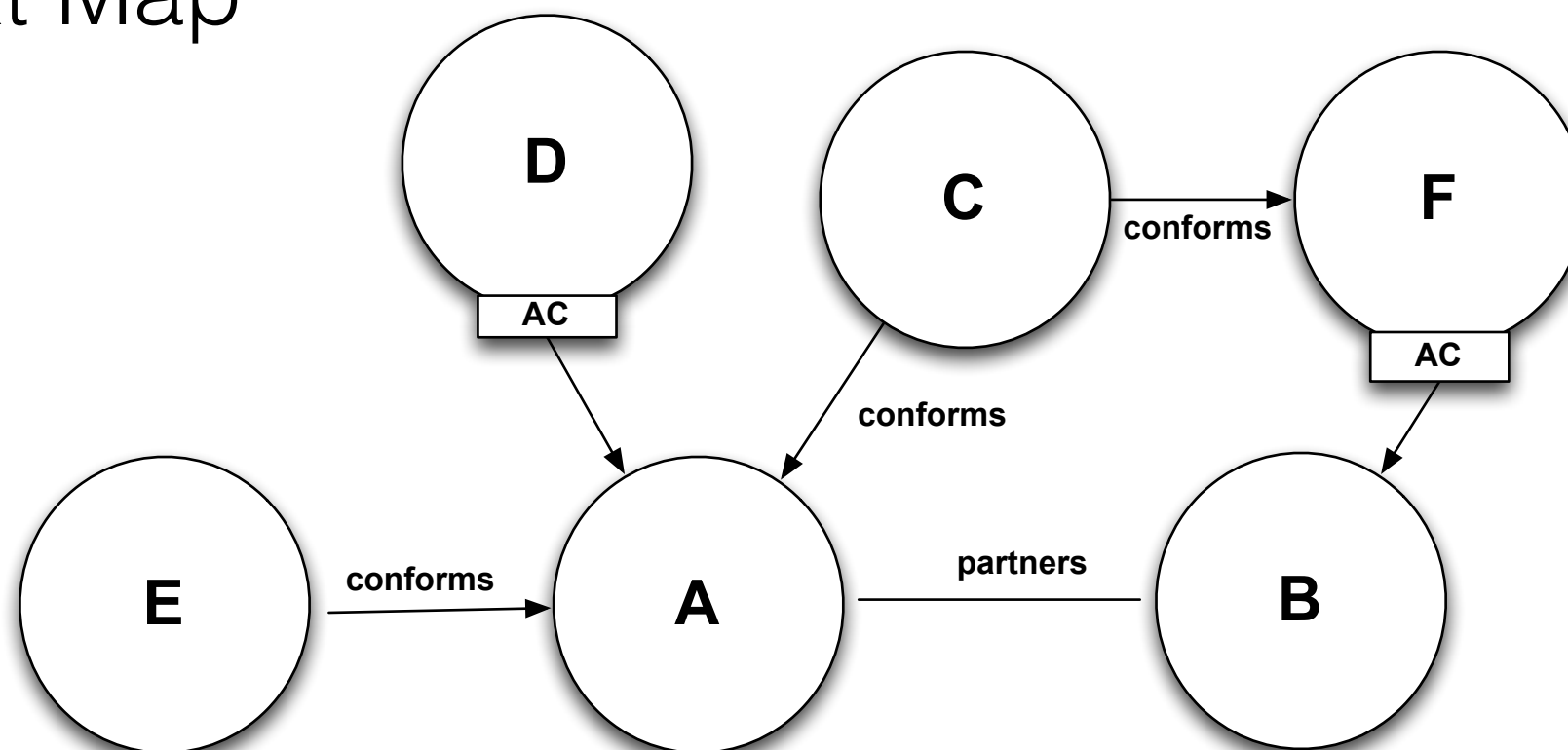
Models need to be clear, not big.

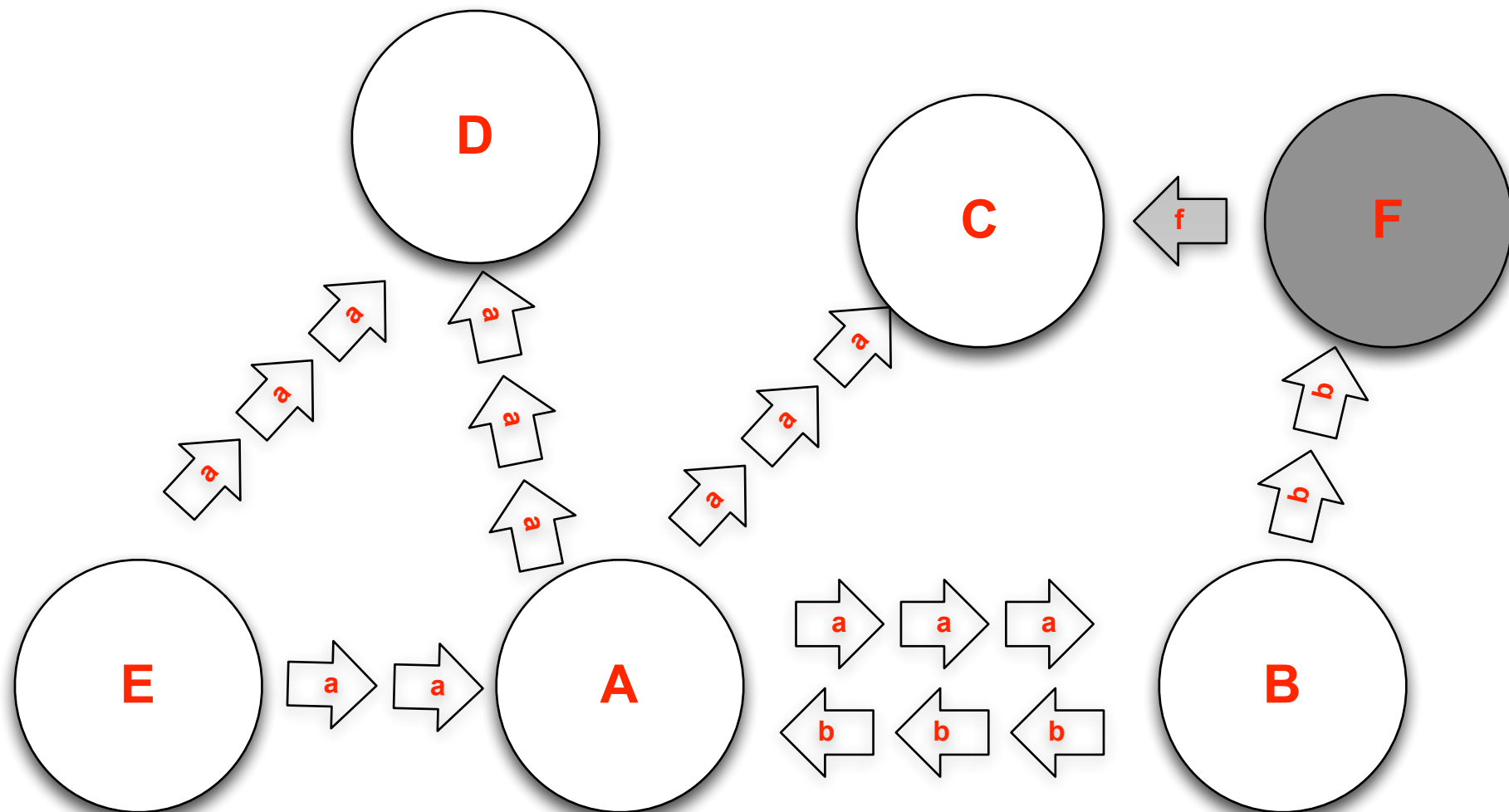
- Useful models need crisp definitions.
- Definitions require clear context.
- Useful models need simple assertions.
- Assertions require boundaries.

There are always
multiple models.

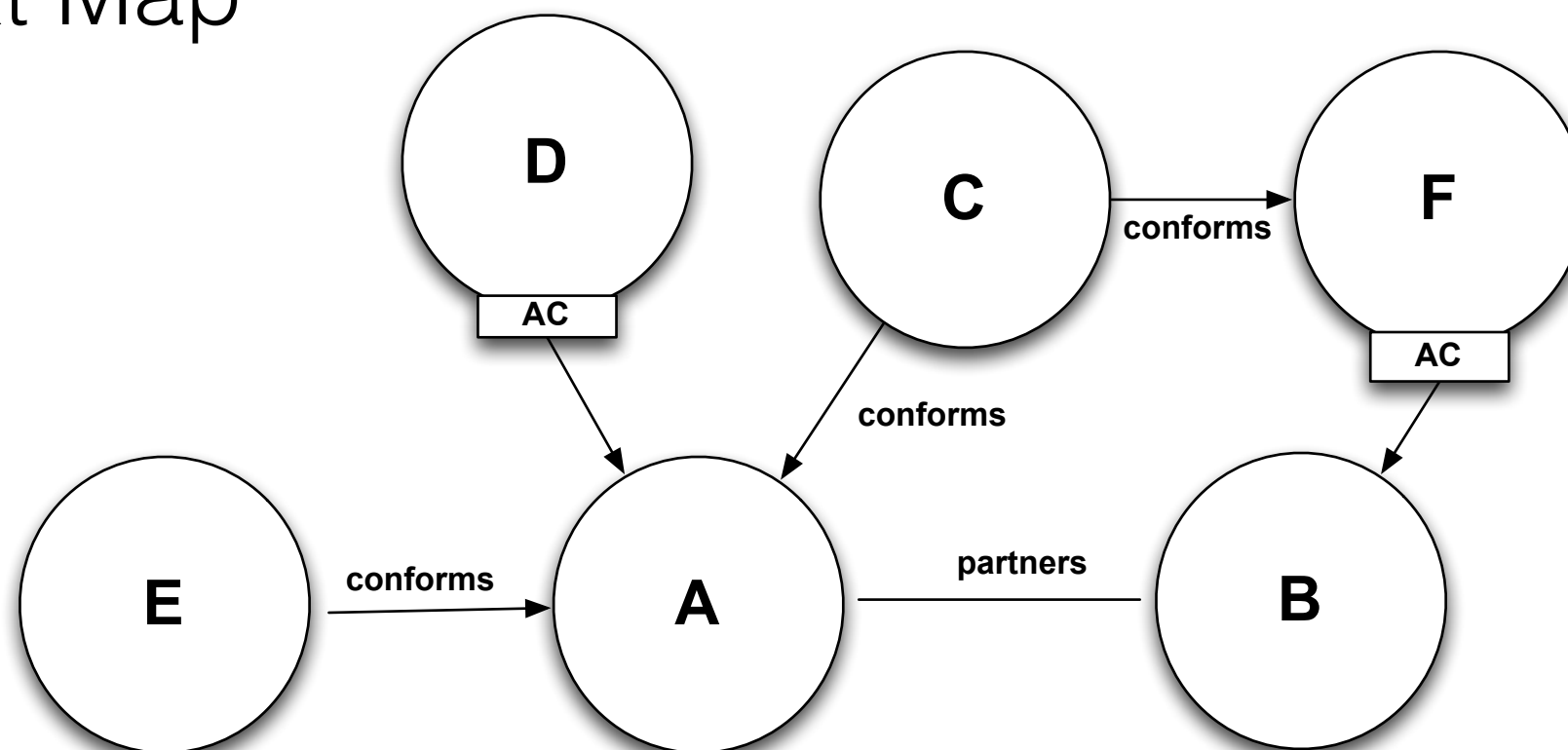


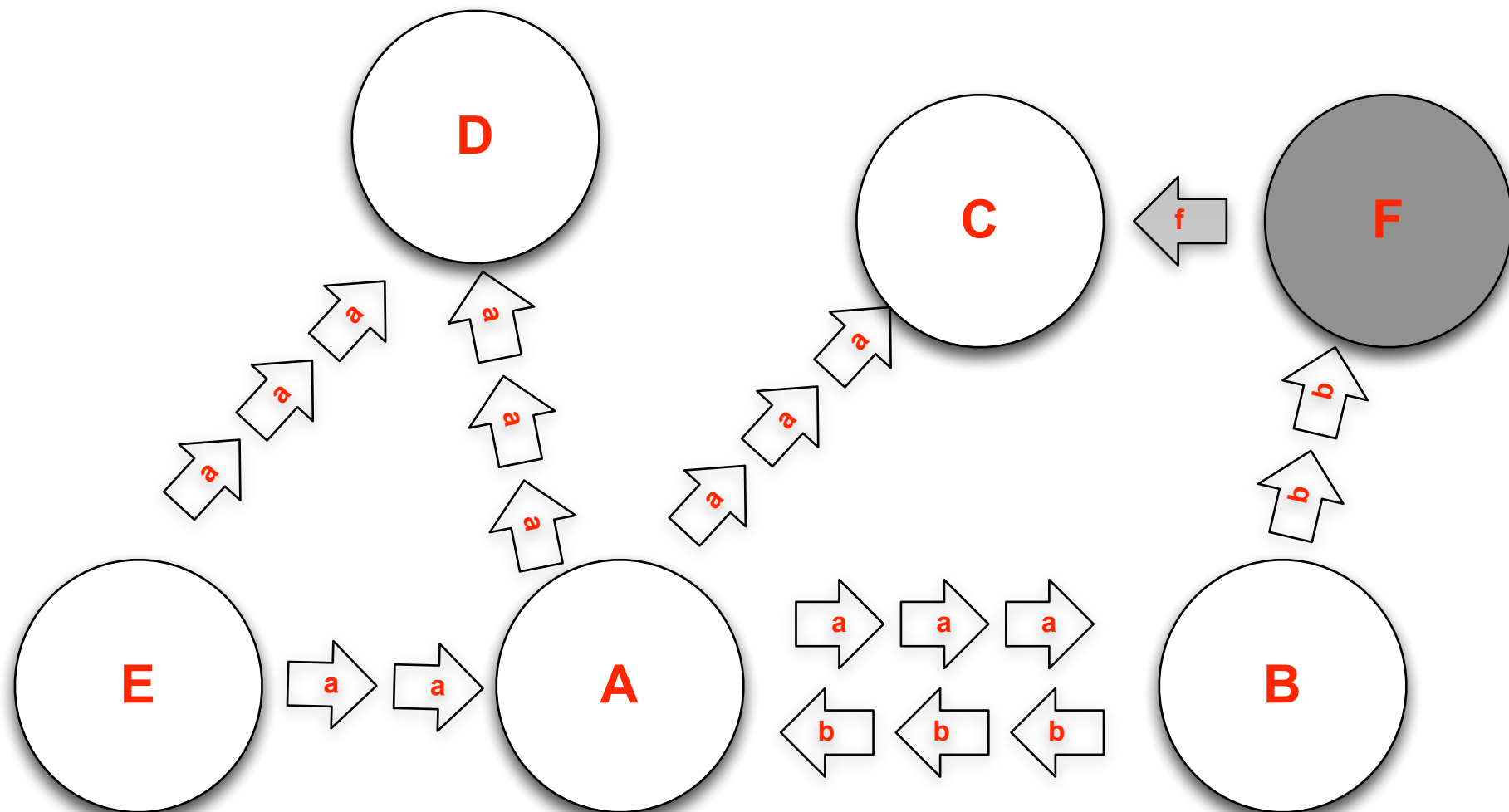
Context Map



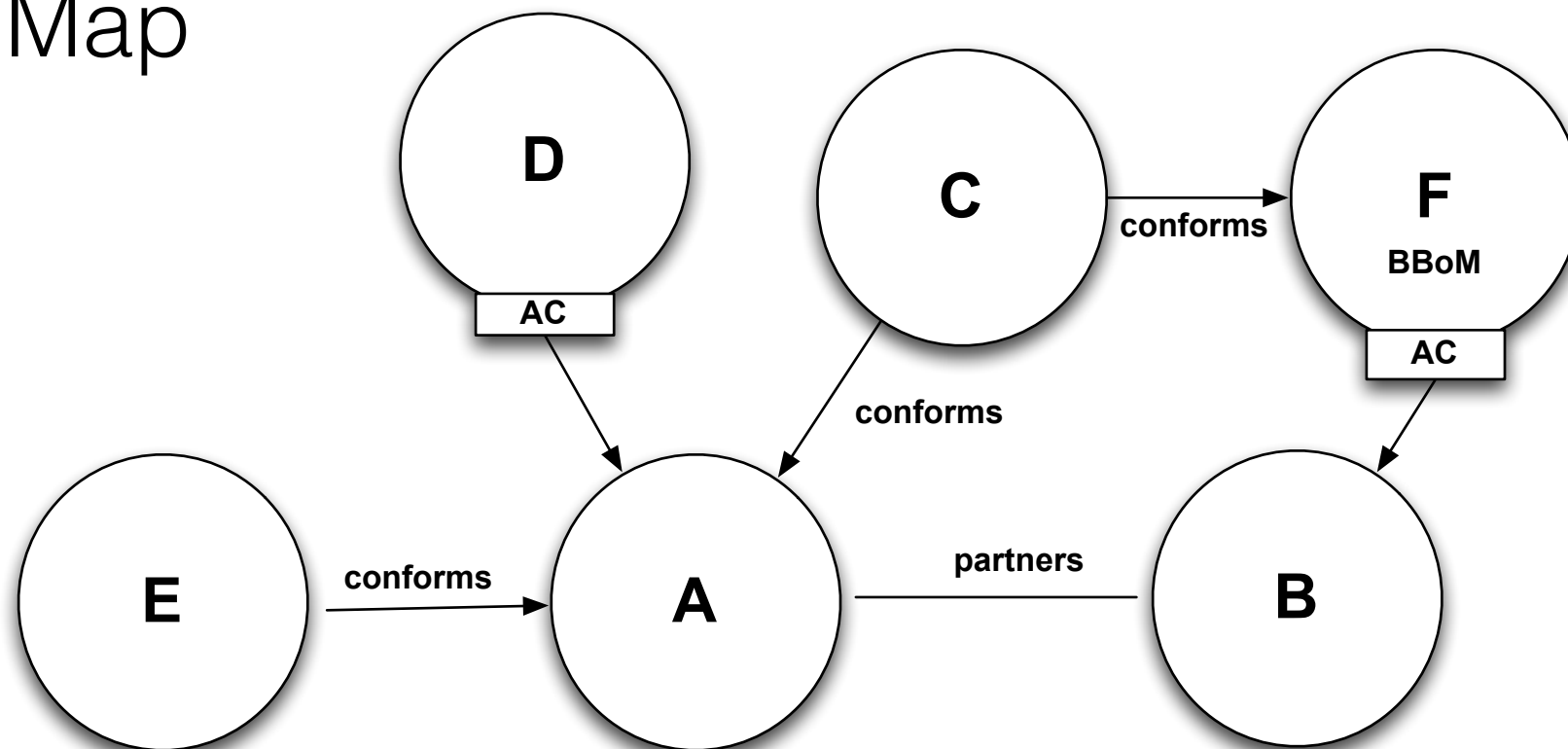


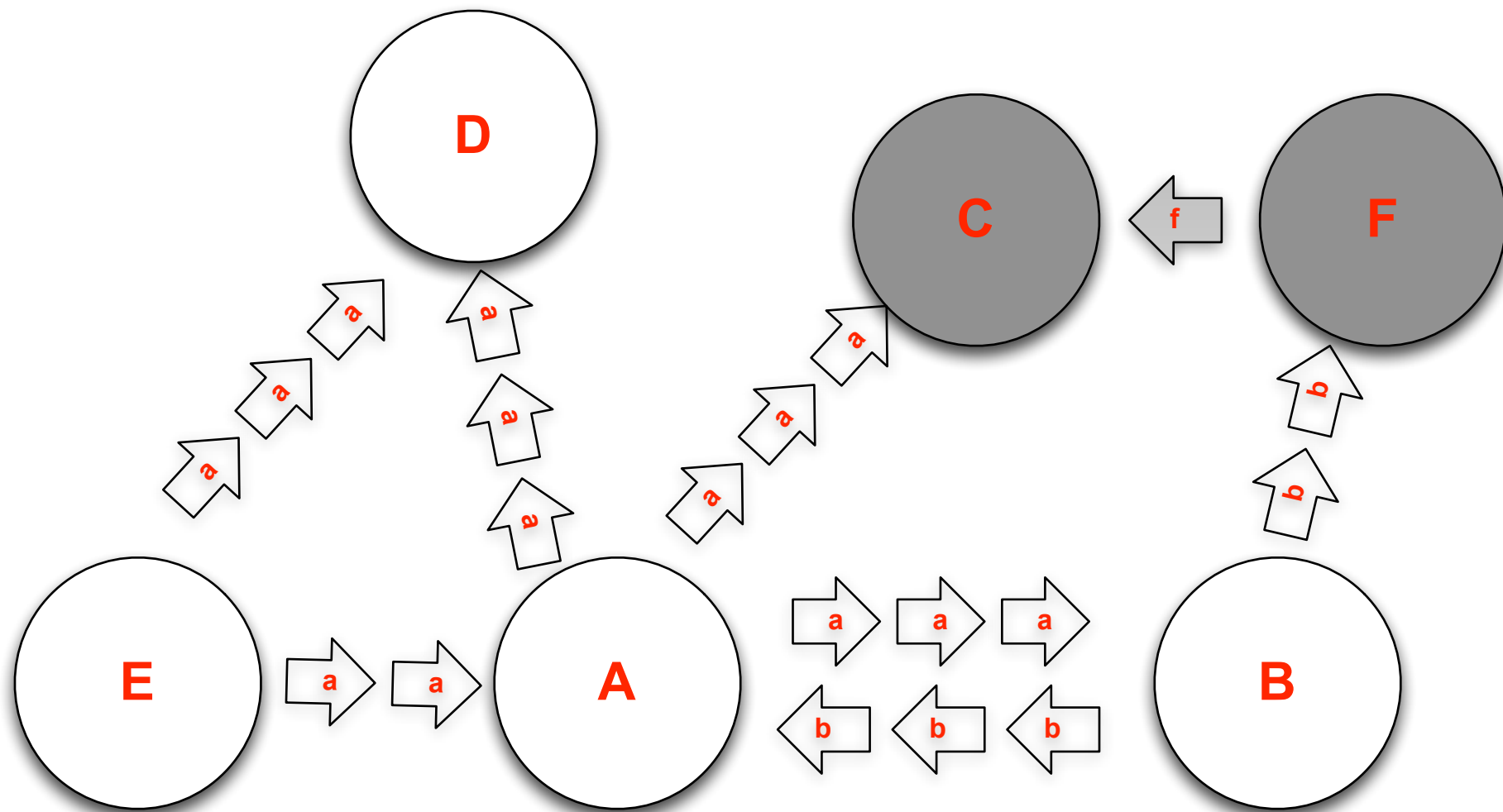
Context Map



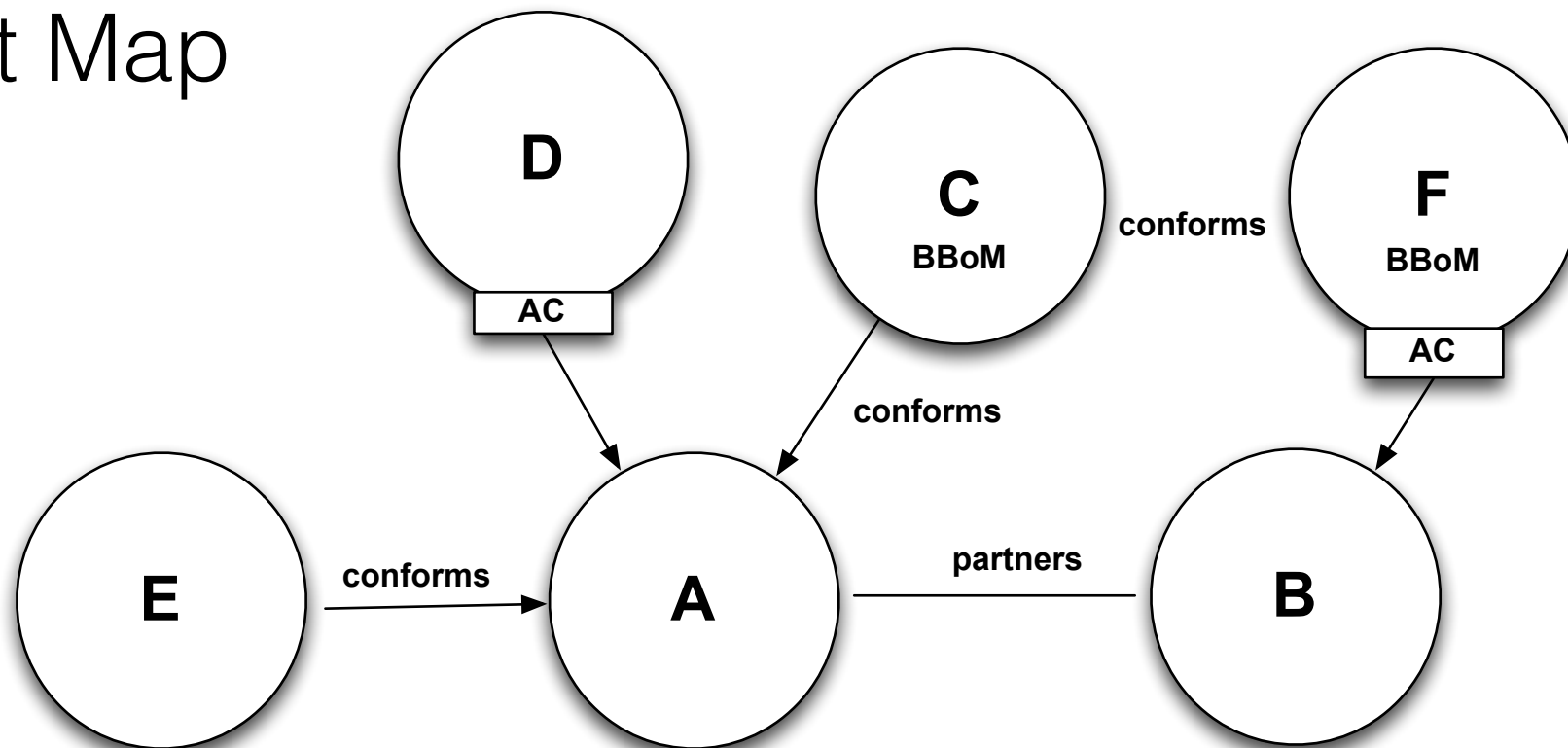


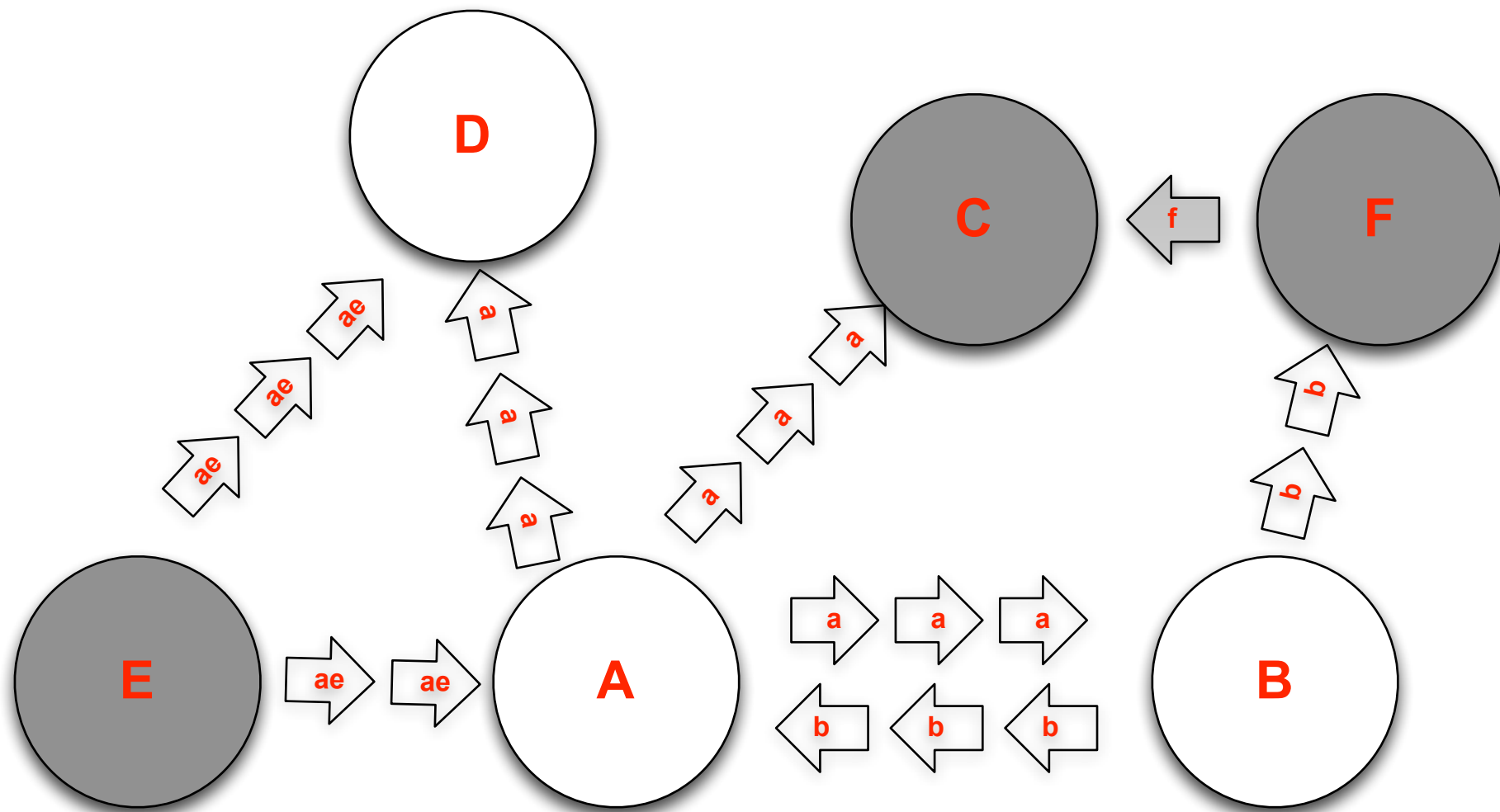
Context Map



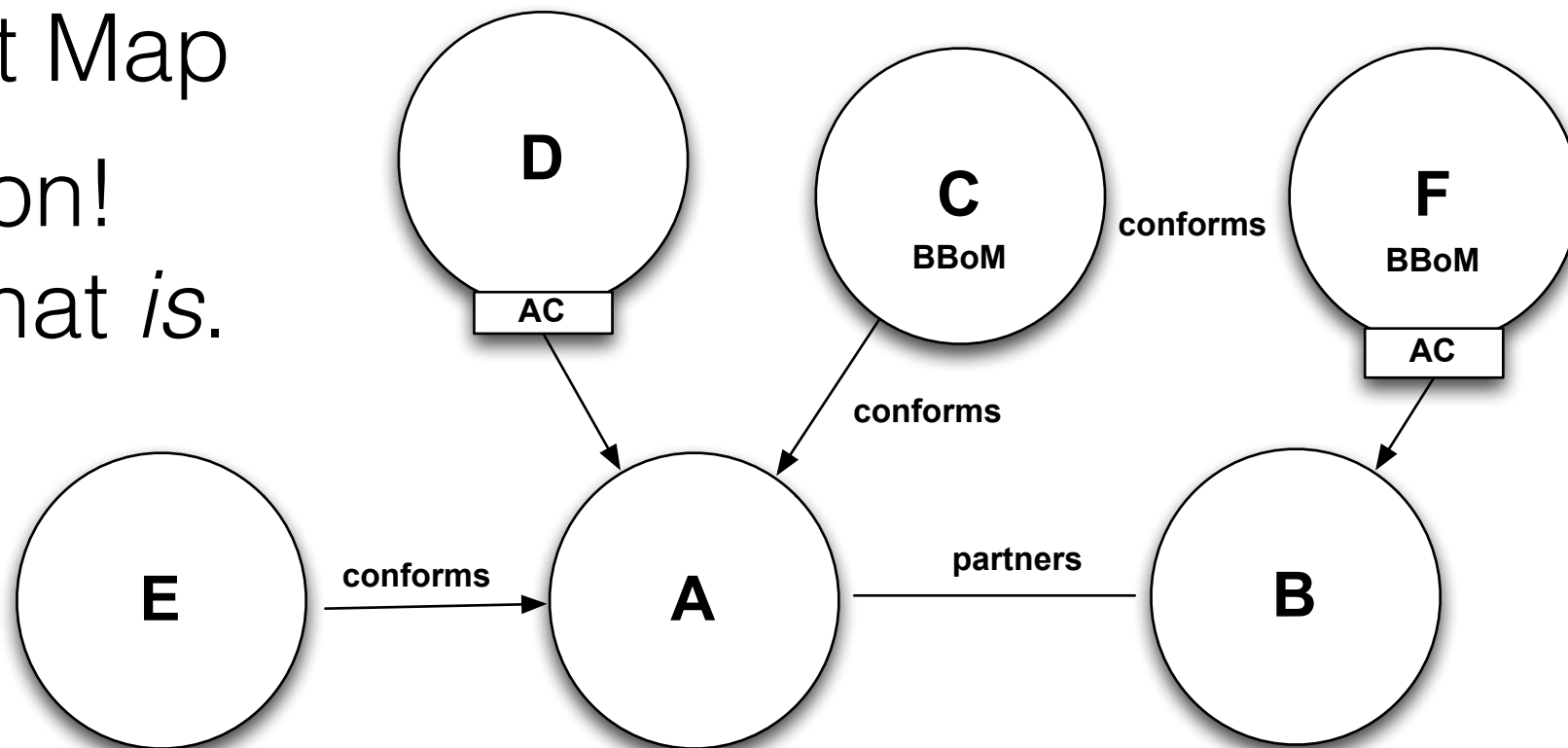


Context Map

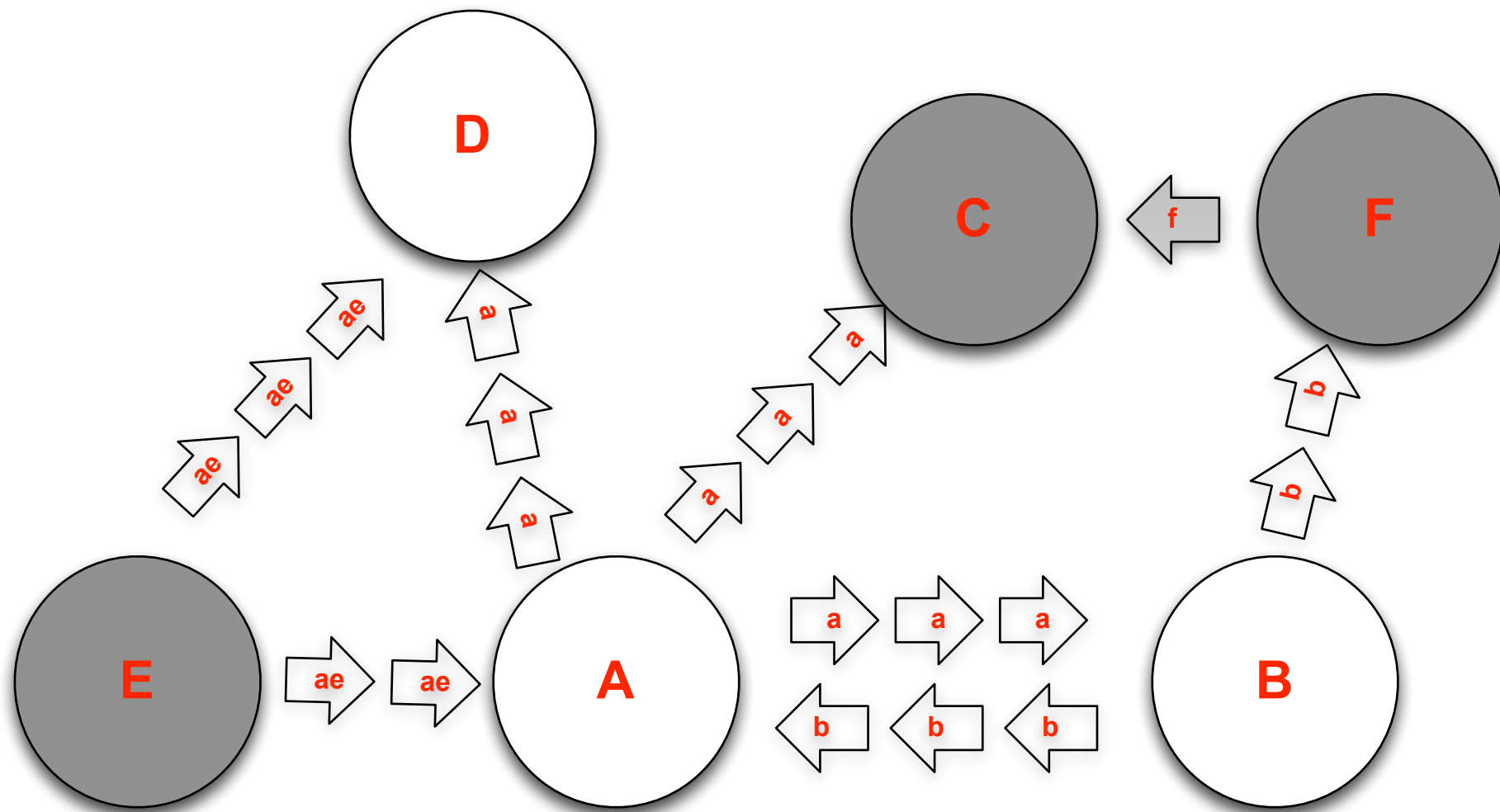




Context Map
Fiction!
Map what *is*.

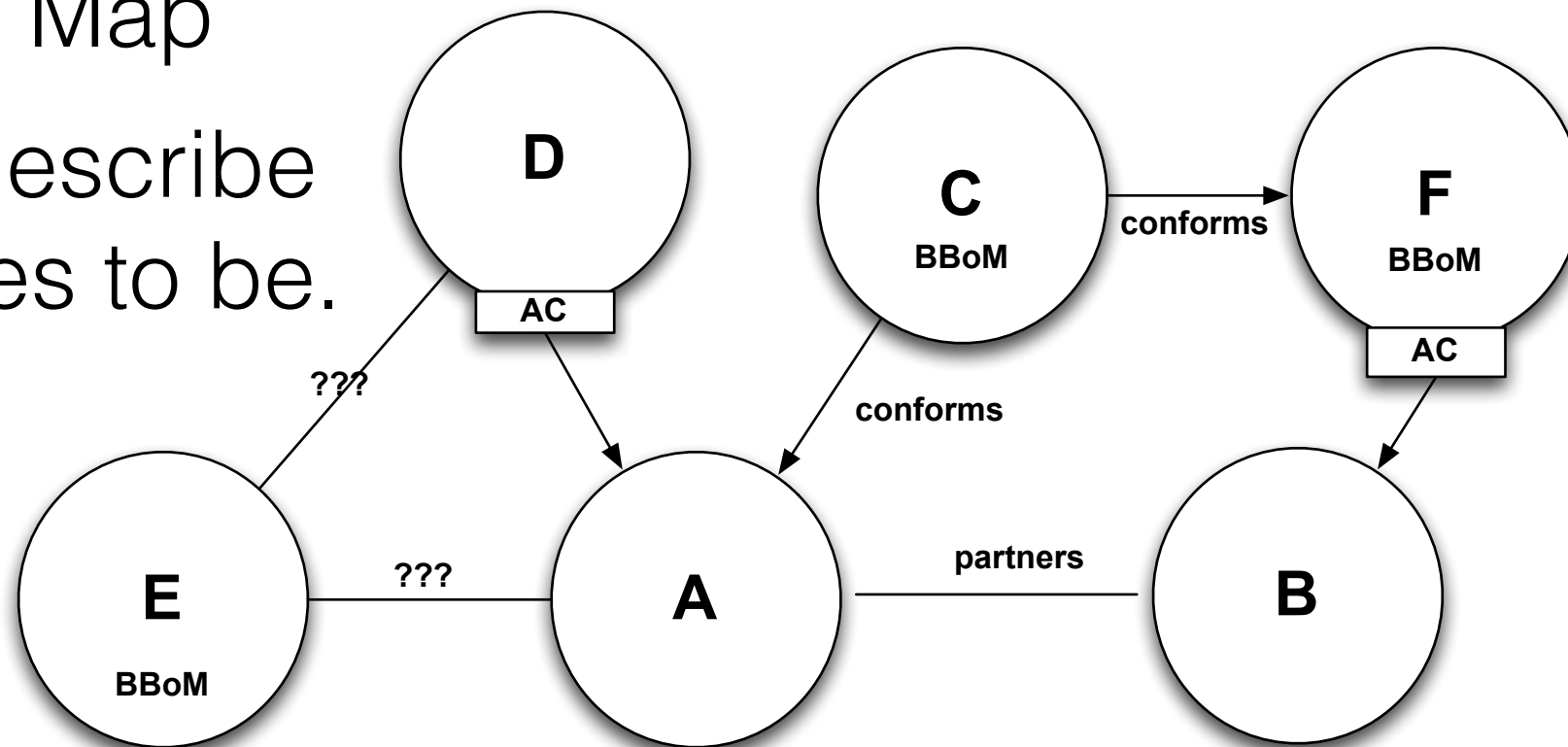


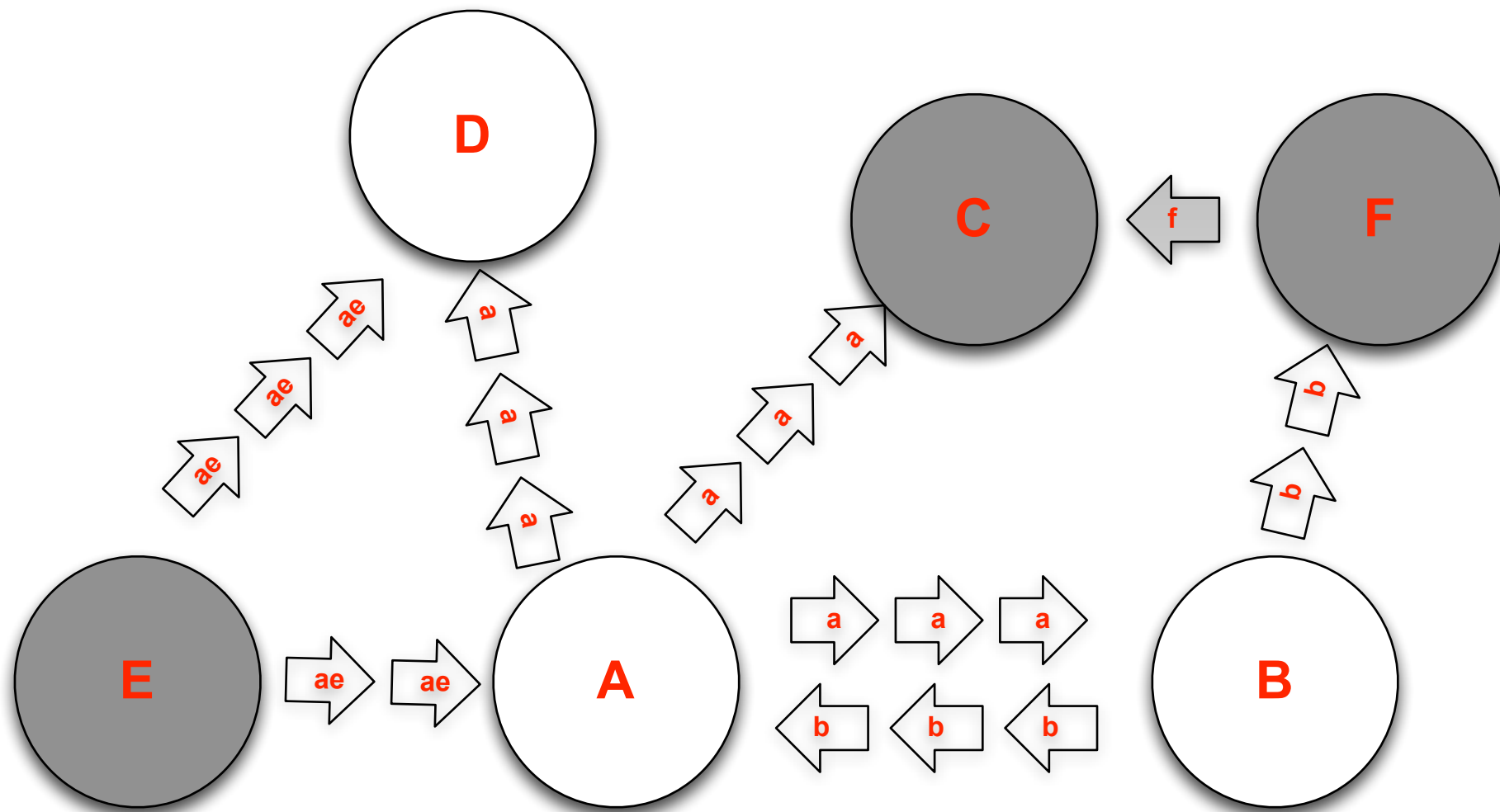
Not all of a large system
will be well designed.



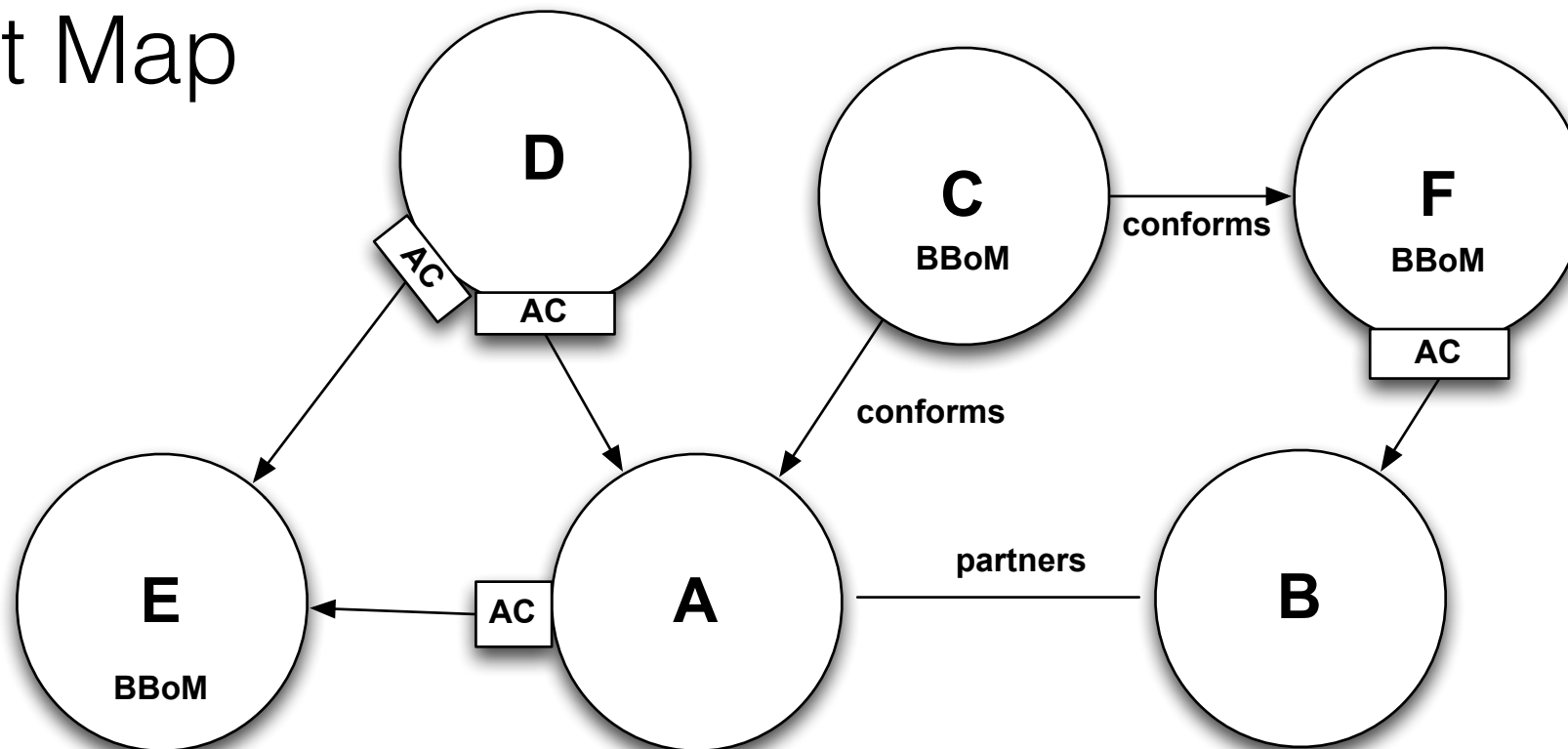
Context Map

Patterns describe good places to be.

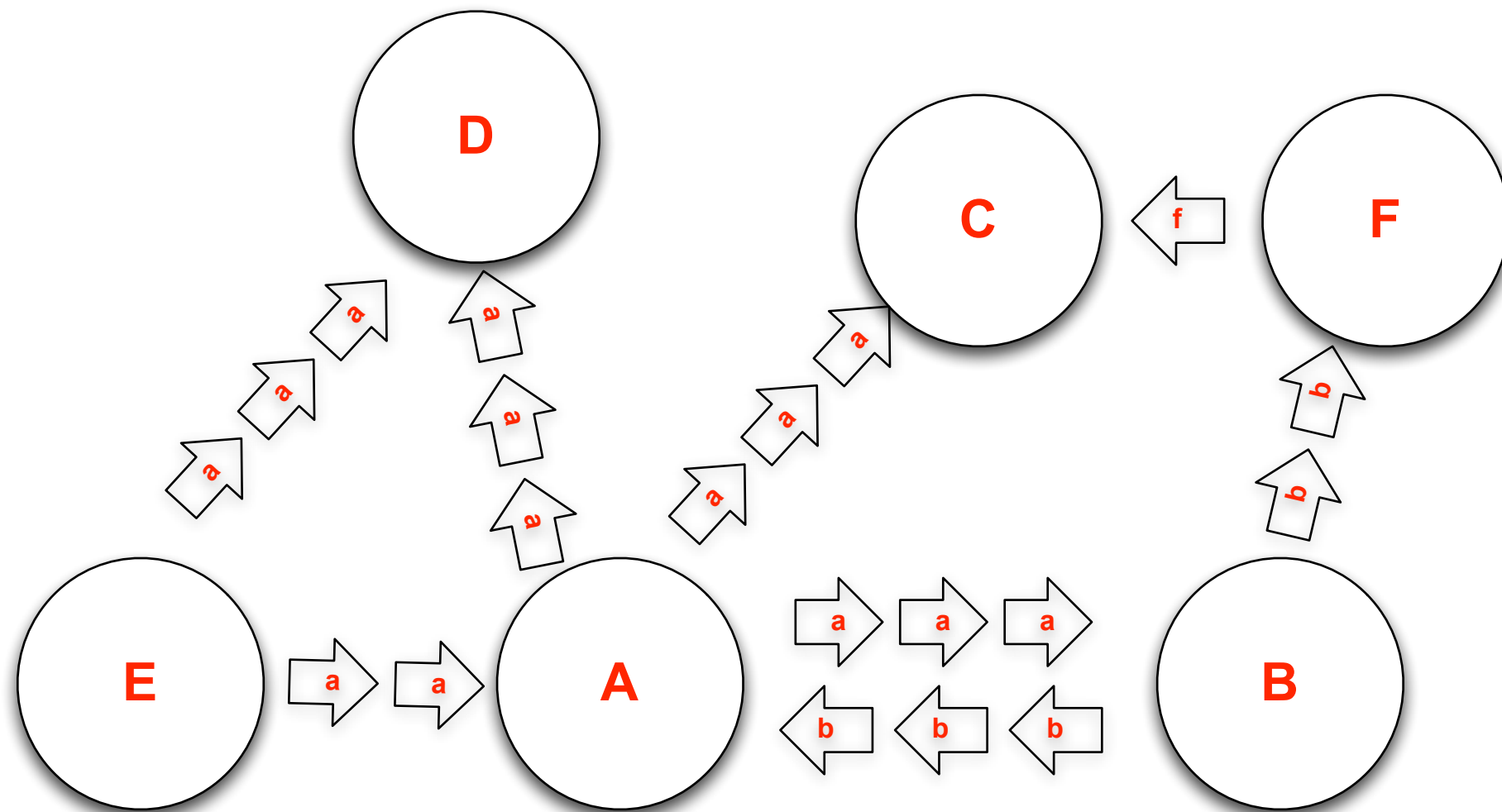




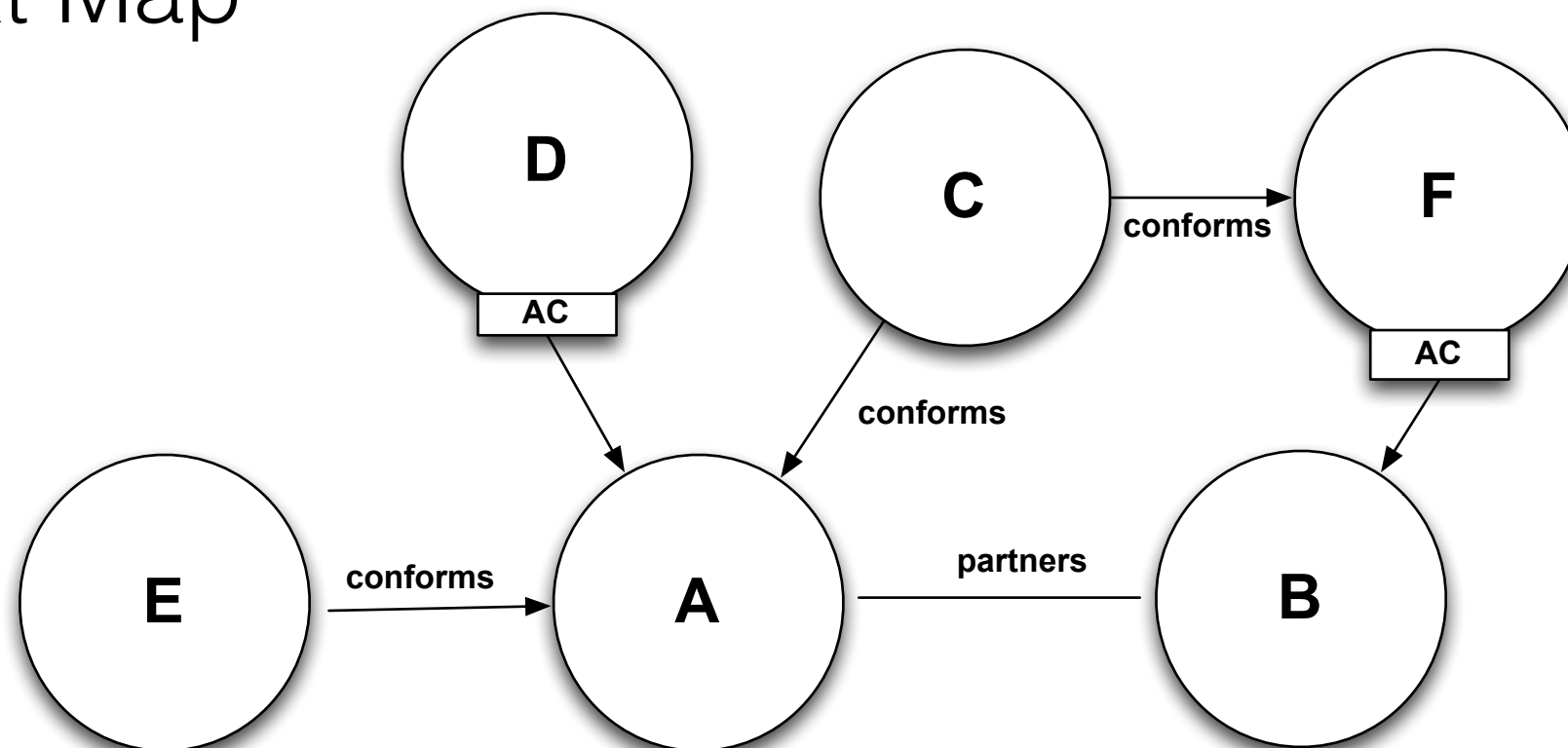
Context Map

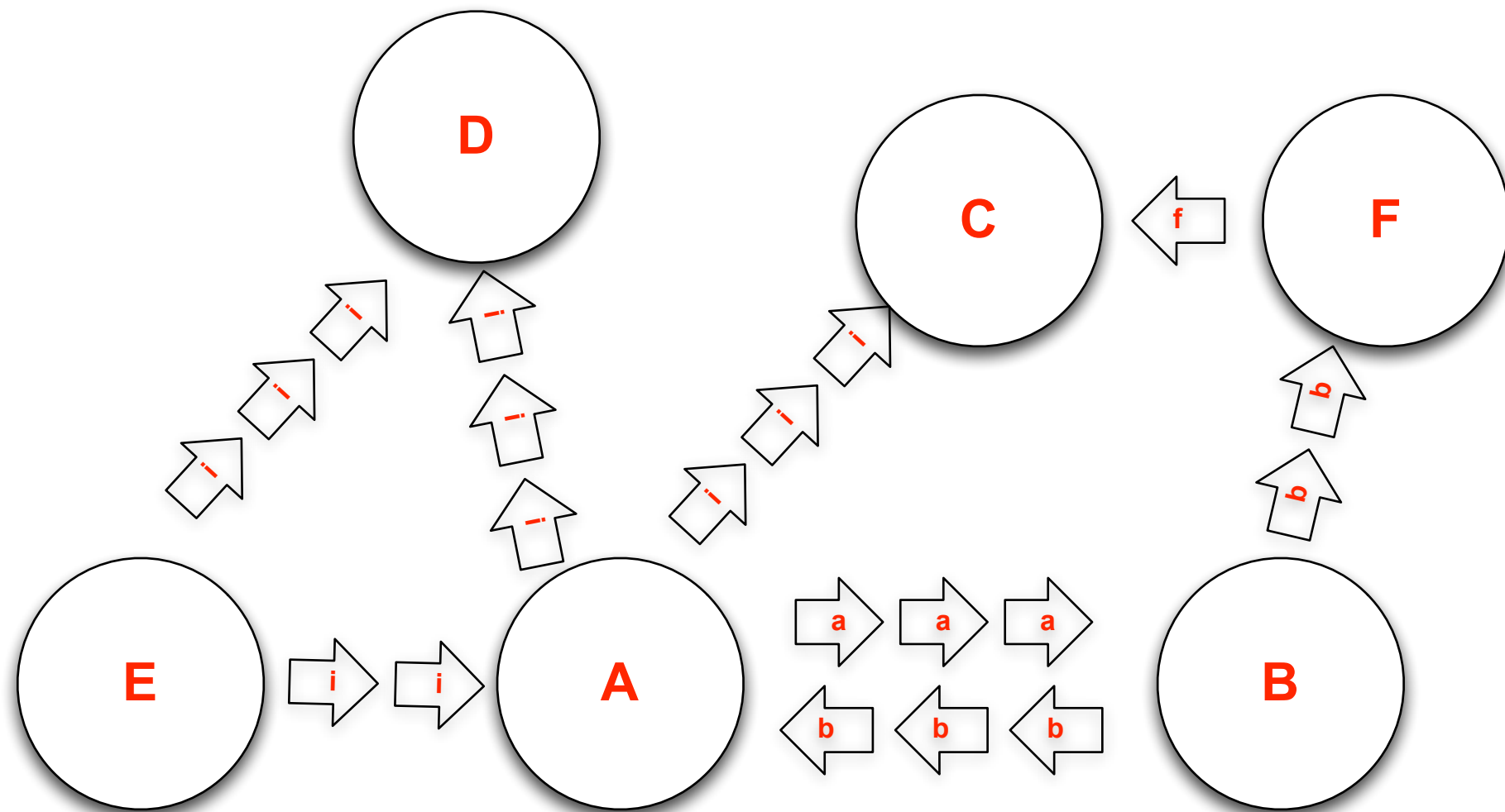


Interchange context

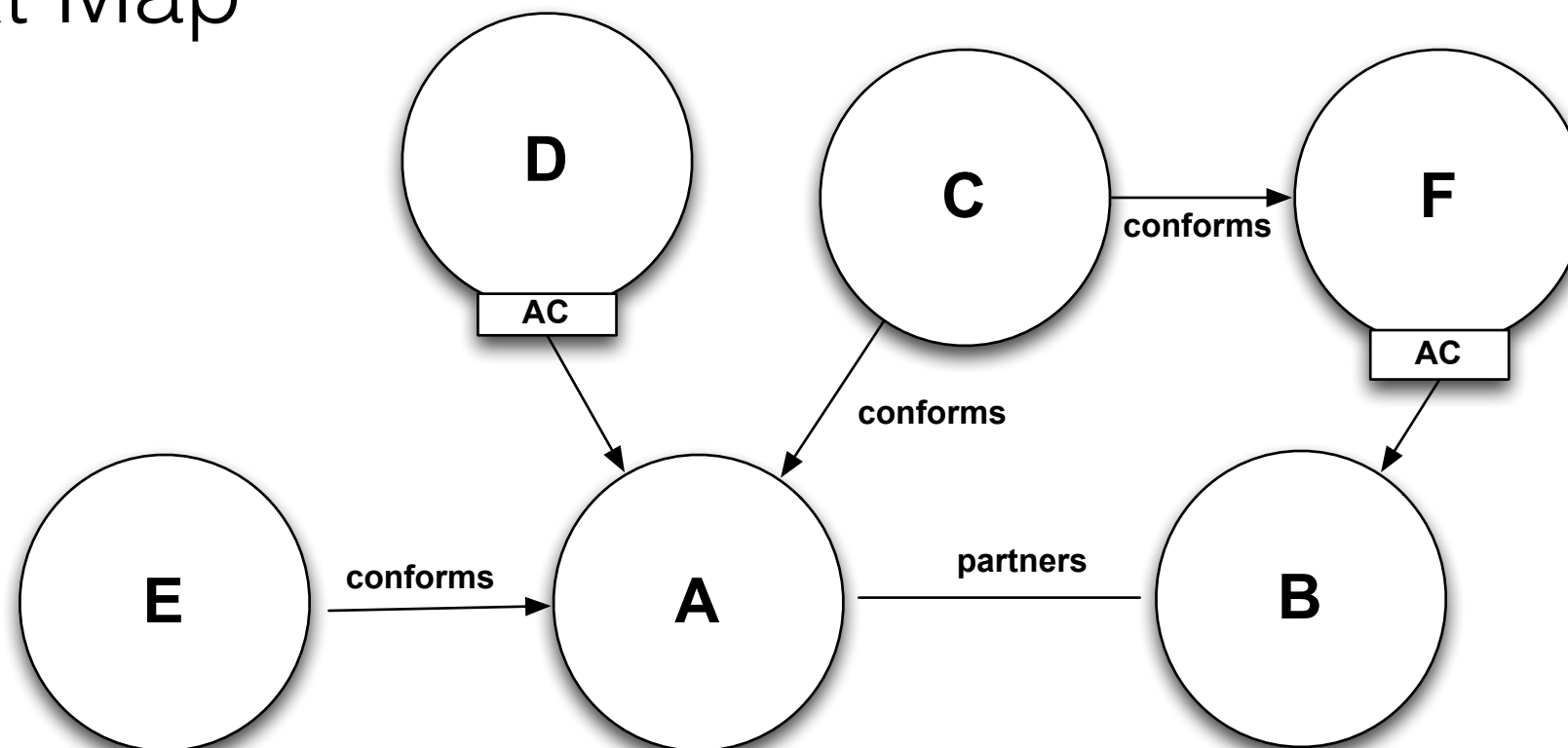


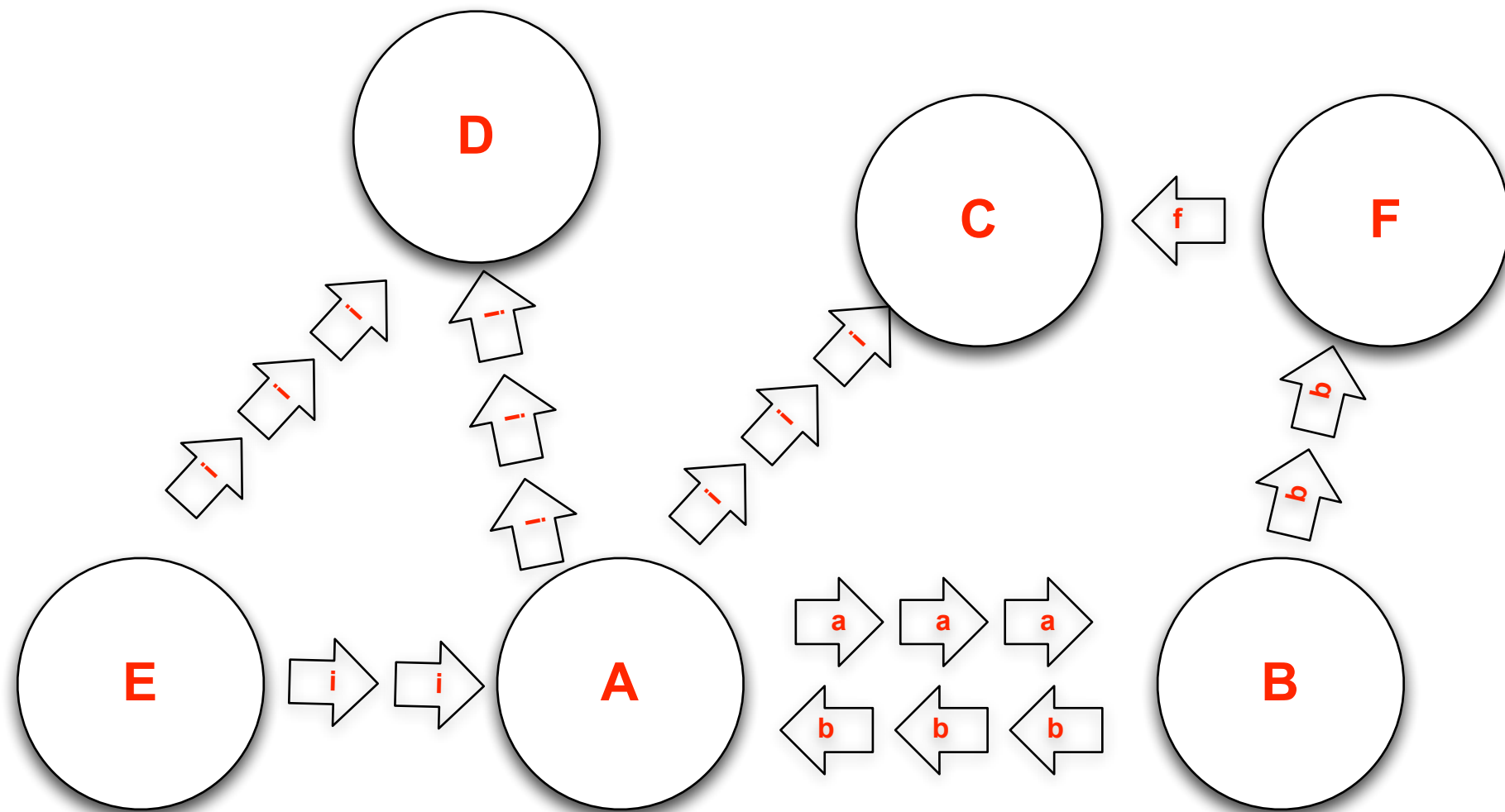
Context Map



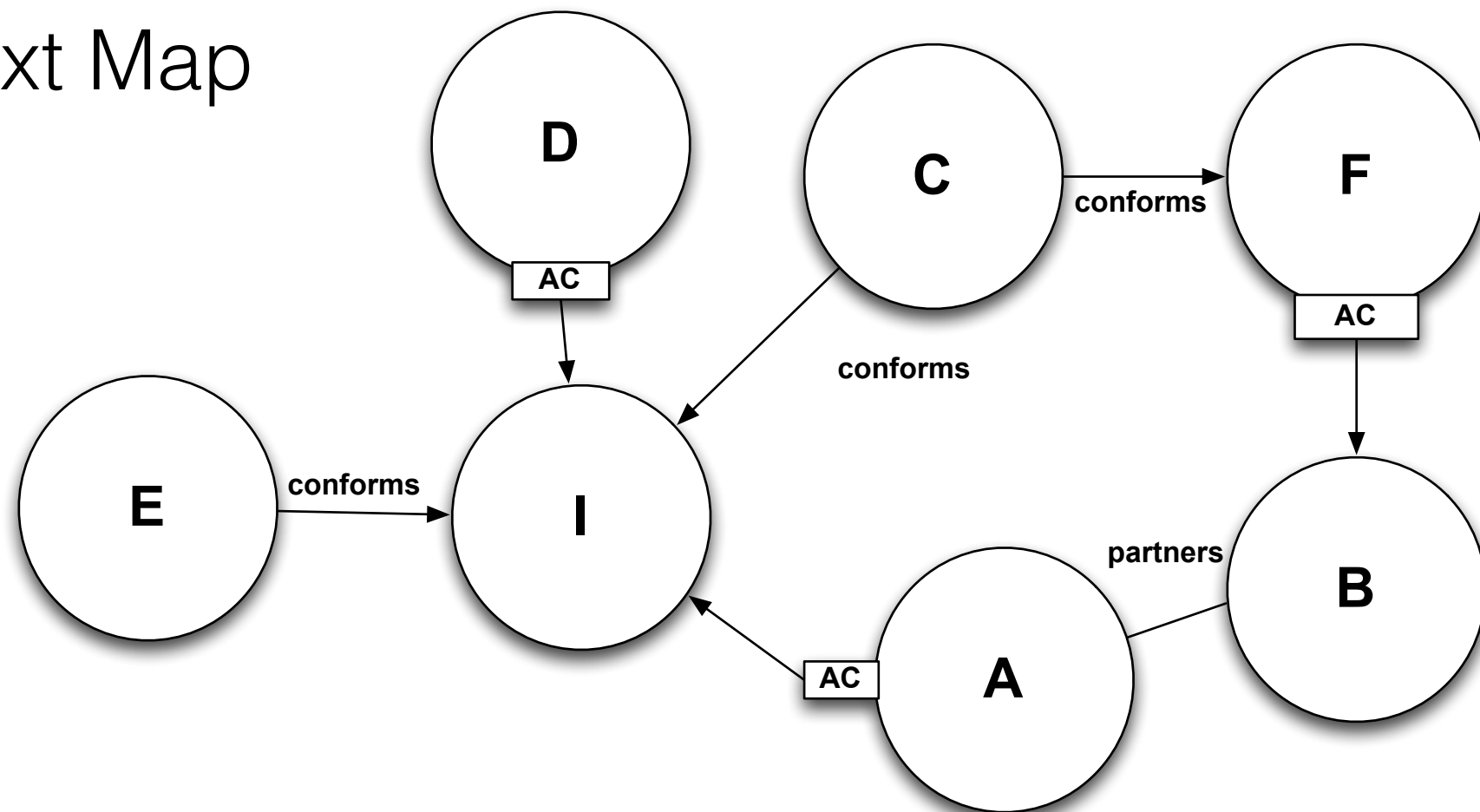


Context Map





Context Map



Why not logical boundaries?

- Smart people point out that most of what I want is the logical partitioning of the system.
- We've had decades to get that to work.
- Some techniques are too subtle to survive the rough and tumble.

Wrap up

- Subtle design (such as DDD) requires concrete boundaries.
- Lots of services recreate some of the old problems.
- Context Maps help visualize and communicate about those problems.
- *Modest* use of interchange contexts is *one way* of organizing coherent services.

Not all of a large system
will be well designed.

DDD & Microservices

At last, some boundaries!

Eric Evans
@ericevans0
domainlanguage.com