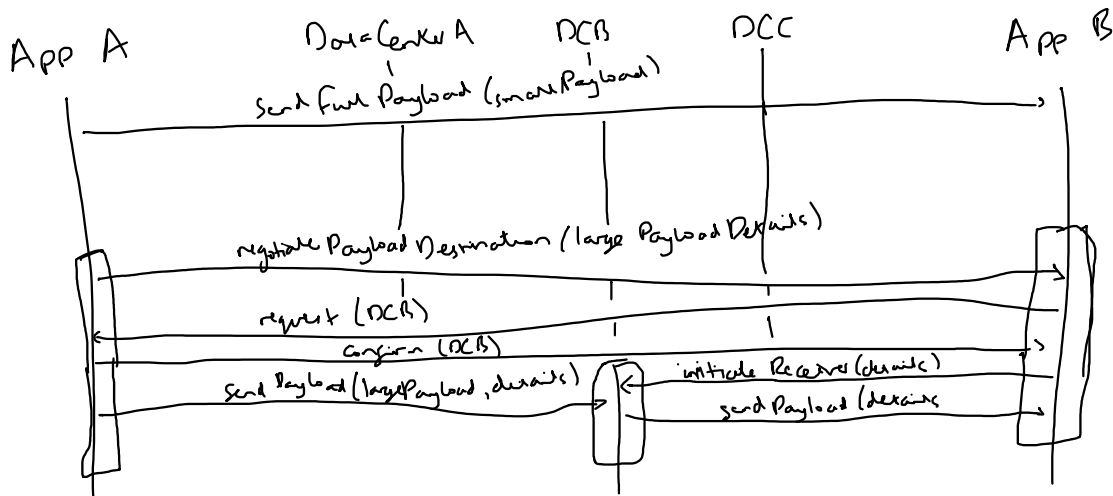


Traditionally, data transfer costs are largely assumed by the sender. Any receiving costs are fixed inbound costs, unless the receiver intentionally updates the payload destination for the actual destination.

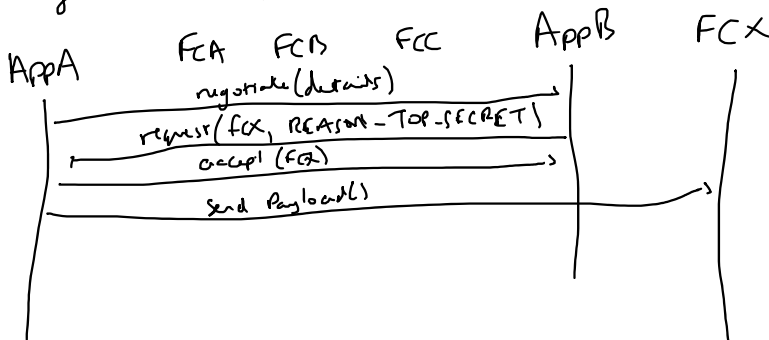
MCPs and the better ability to span remote processes should enable "fairer" transfer decisions. Instead of forcing one party to assume the entire cost, the sender can negotiate with the receiver where they will send a sub-receiver to catch the payload before finishing the journey for the receiver.



This assumes the priority is balancing distance / cost.

Payload fulfillment centers can also state locations and reasons, where requests get sanity checked (either help checked or reject checked).

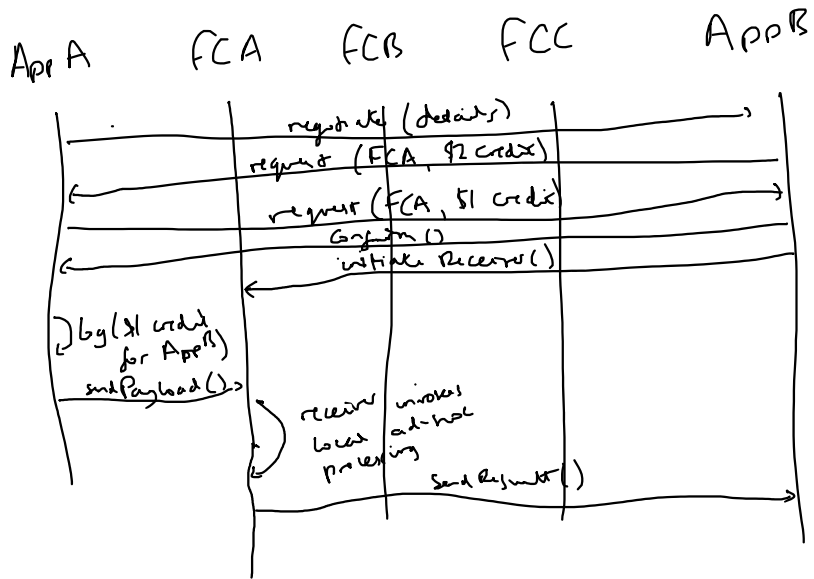
For example, if AppB needs a more secure destination they may choose somewhere that is even further than AppB itself.



Each application will best know how important the raw data is, or how much could be processed and dropped pre-transmission.

In the age of serverless ad-hoc computing, an app could request a datacenter closer to the origin in exchange for some good-faith credit. That could be negotiations too - given the payload is assured costly enough to warrant negotiations, back and forth with this amount of data should not be

closer to the origin - given the payload is assured costly enough negotiations back and forth with this amount of data should not be problematic.



This creates a situation that is advantageous for the origin, destination, and overall resource distribution.

Given wireless resource demand variation, it's possible that the destination will vary based on real time availability, and the negotiation will include multiple options.

