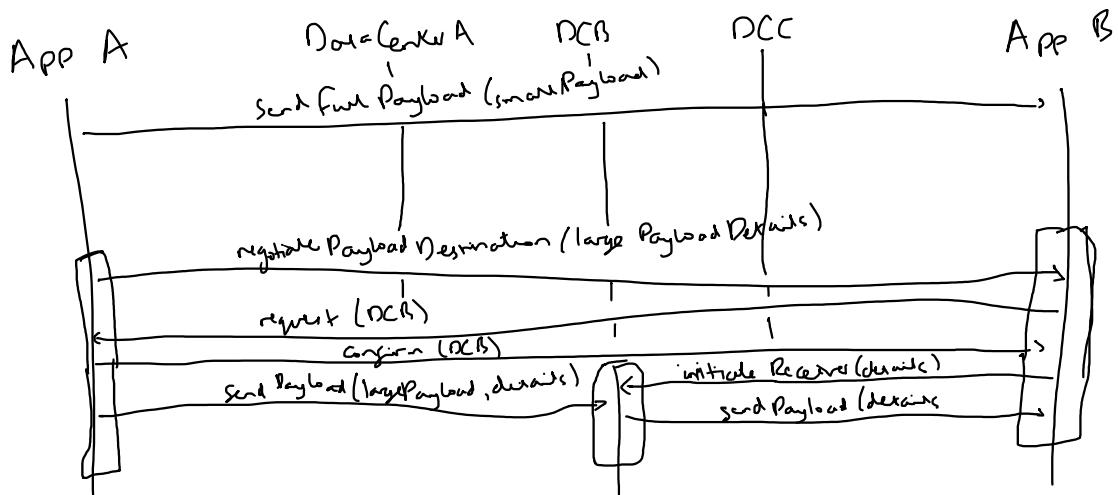


Payload Receiver

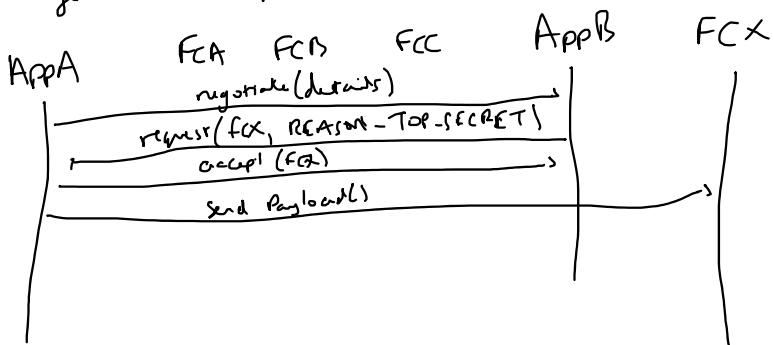
Friday, January 30, 2026 3:02 AM

Traditionally, data transfer costs are largely assumed by the sender. Any receiving costs are fixed inbound costs, unless the receiver incrementally upgrades the payload destination from 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 other receiver nodes who will lend a sub-receiver to carry the payload before finishing the journey for the receiver.

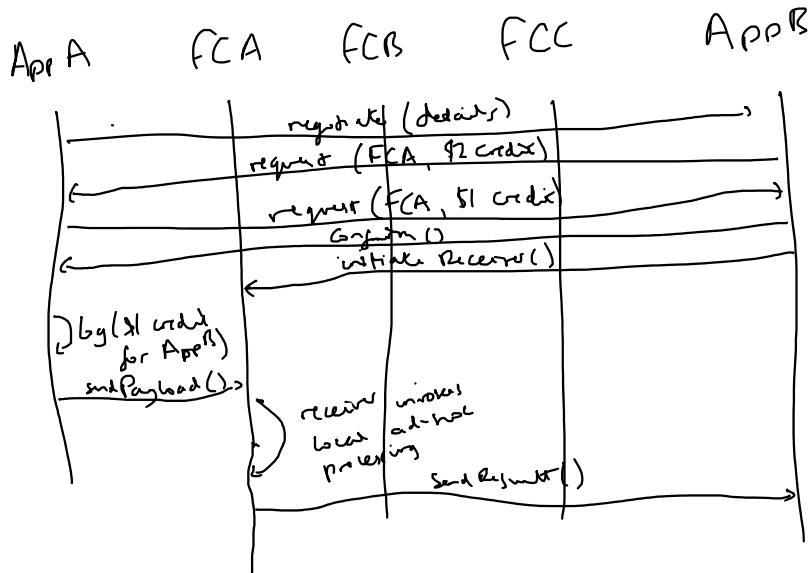


This assumes the priority is balancing distance / cost.
 Payload fulfillment centers can also state locations and reasons, where requests get sanity checked (either self checked or peer checked).
 For example, if AppB needs a more secure destination they may choose something that is even further than AppB itself.



Each application will have known how important the raw data is vs 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 an exchange for some good-faith credit. This could be negotiations too - given the payload is assumed costly enough to warrant back and forth with how much data should not be negotiated.

Clear to see why given the payload is assumed costly enough negotiations back and forth with large amounts of data should not be problematic.



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

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

