# Old vs New

Previously, it was standard to conform to a request/response, and maybe async too. With async it was only "not yet" or "ready", no "X%".

Difficulties handling communications (or just a lack of want) meant that nested calls introduce huge uncertainty, hence can avoiding using things like promises too much.

There's no reason it shouldn't enable an interactive object being passed through. There are security implications, but the utility should outweigh any rigid concerns that can be handled anyway.

Instead of initiating a request, pass an object with outbound communication enabled and optionally inbound communication too.
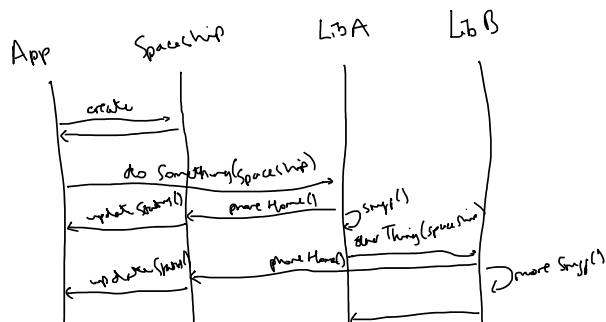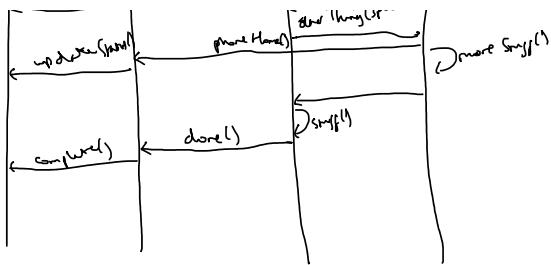
Before: Now
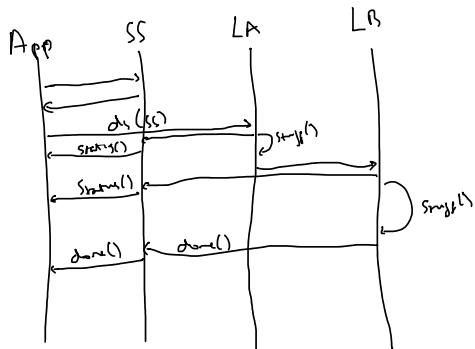
This enables:
- in transit status updates
- Optimum payload size (minimum payload first)
- dynamic routing post transit
- efficient return routing
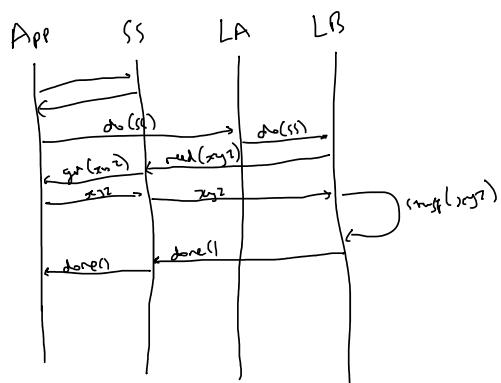
Notice the call home on entering any foreign library.
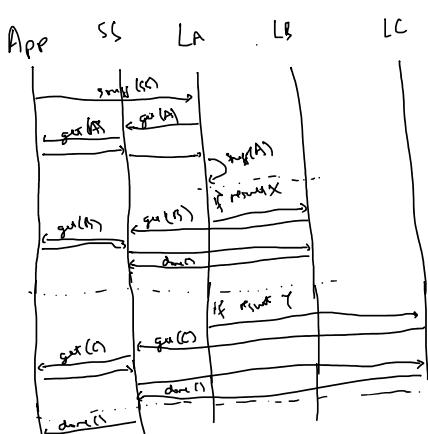
If there is a flag to determine whether the spaceship must return before going home, that enables skipping libraries on the way back.
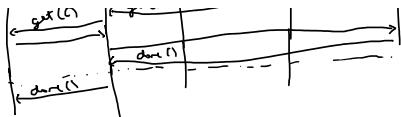
MCPs have introduced the concept of "I have this, do that". This should also enable "To do that, I am assigning this".
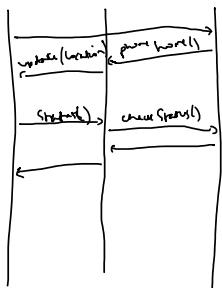

This also enables "down the line" conditional branching with branch specific payloads.
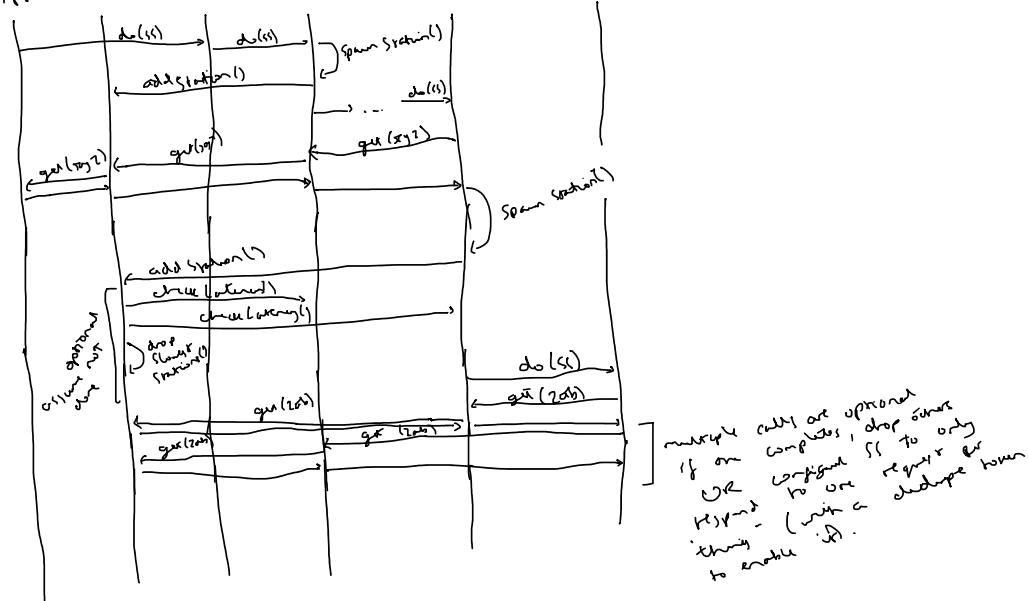
The fringe benefit of a spaceship calling home is that it presents the opportunity for any inbound connectivity too. The outbound phone home can include requests about how to call back.
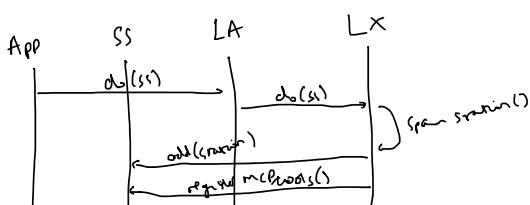
MCPs generalise spawning processes in remote locations. This means that spaceships can also "drop" docks as needed — either for their own return purposes, or to mediate calls home in an effort to improve latency. The only additional requirement is to ensure all docks are cleaned up before concluding execution. Some default timeout should help ensure this.
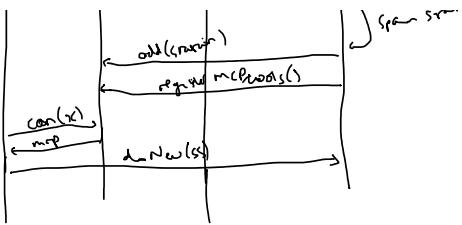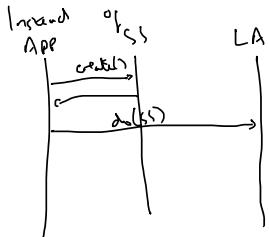
The space station can also enable new "native" calls for the origin app. This is increasing available granularity in line with whatever has already been used at least once.

Diagram labels: add(creator), regulated mcp tools(), con(x), mcp, do New(ss), Span Sym
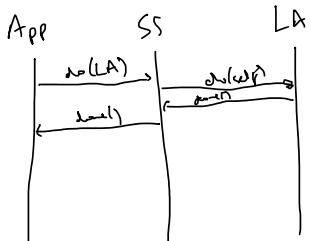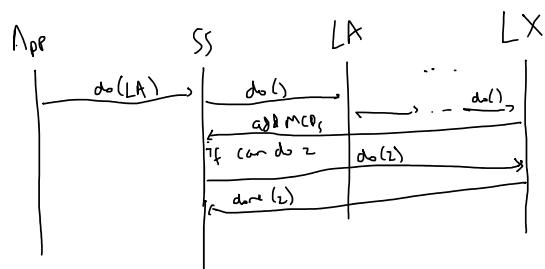
This suggest we update the model slightly so that the comms interface is a shuttle carrying the payload with it.

Instead of SS

App   LA

create()
do(ss)

It would be

App   SS   LA

do(LA)   do(ss)
done()   done()
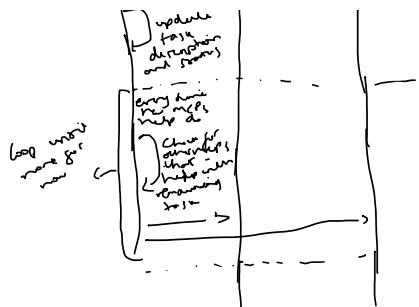
As well as enabling tools for later use, the spaceship may be carrying some known "need to dos".

App   SS   LA   LX

do(LA)   do()   ... do()
          add MCPs
          if can do 2   do(2)
          done (2)

This would enable semi intelligent space ships to self navigate using a predefined task that does not have any predefined execution. The requirement would be either some "onboard" interpretation engine, access to a local engine, or access to a separate detached engine.

App   SS   LA   Lx

do(LA)   do()   do()
          requires mcps
          do any mcps help complete closs dirunption
          if yes doOne()
          do Tool()
          update task dirunption and scores
          carry on ...

Loop until done 8° now?

- update task description and status
- every time recurse new d?
- Check for ownership that will help with executing task

This enables long lived, widespread task execution.