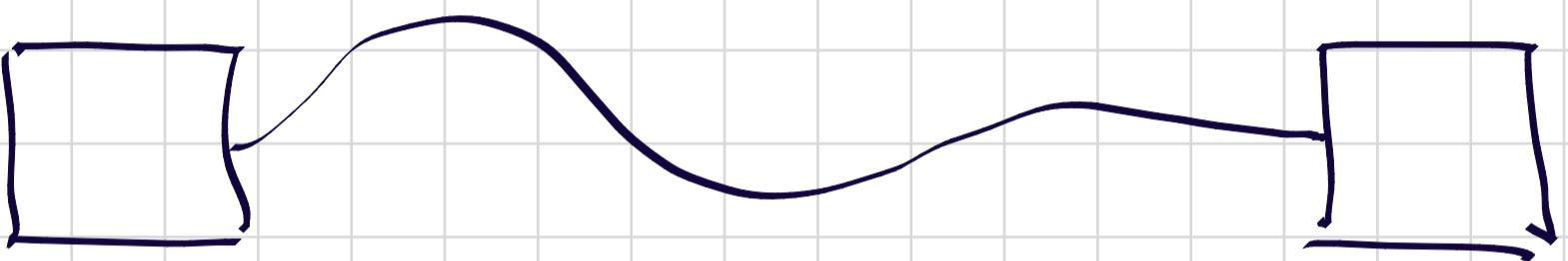


Create a bike path. A path starts at a space and ends at a space. It could have many spaces in the middle.

We can use a space factory - to create intermediate spaces.



Two pre-defined spaces connected by a path.

JSON object



x = create space attribs-s

y = create space attribs-s

other attributes



attribs-s = {loc: [x,y], size: [w,h], type: —, ...}

p = create path attribs-p

attribs-p = {start: x, end: y}

Let's consider a complicated situation. Here, we have a much longer path. We want the path to be well defined so we can help Space OS in its path \rightarrow trail mapping.

To constrain the path (i.e., to specify the shape in little more detail), we introduce waypoints. Waypoints can be introduced in a path in two different ways: (a) specifying an array of Space instances, and (b) specifying a Space factory that produces the Space instances.

Space instances have definite locations and sizes. Space factories can be parameterized to construct Space instances that have specific characteristics. For instance, we can generate spaces in a defined trajectory.

p = create path attrs-p

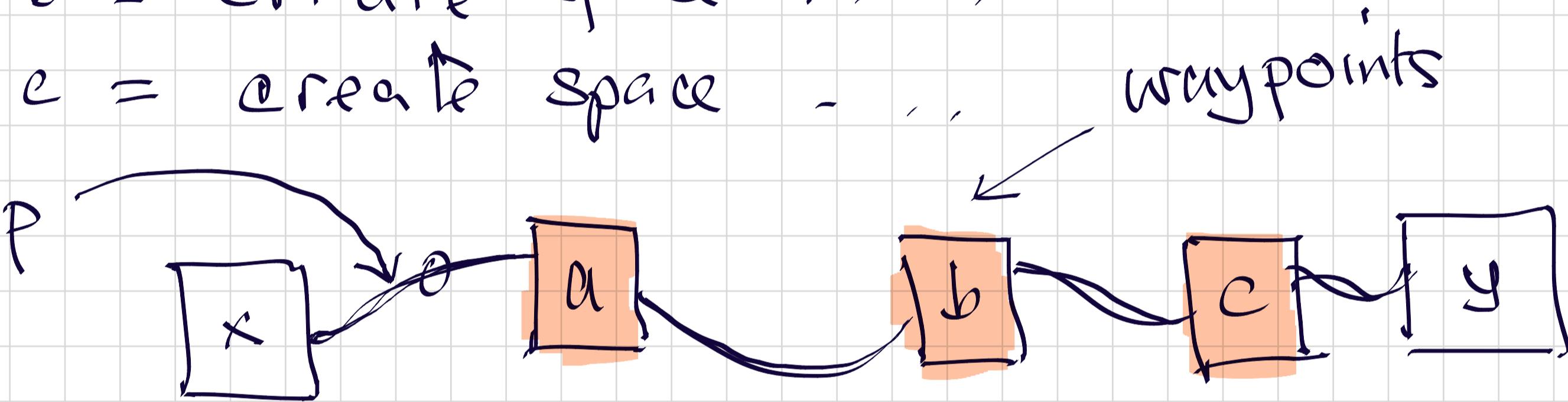
x, y : already defined.

{ start: x, end: y, waypoints: [a, b, c], ... }

a = create space attrs-s

b = create space . - .

c = create space - . .



q = create path attrs-p

attrb-p = { start: x, end: y, waypoints: z, ... }

factory



factory z is going to create the waypoints automatically
as needed

We can constrain the factory so it generates the necessary waypoints. Here are some example ways the space instances can be constrained.

- spaces in a line.
- spaces of a given size and type
- spaces in a line, starting, and stopping at given points.
- some other ways constraining the locations, sizes of spaces. All spaces are of a given type.

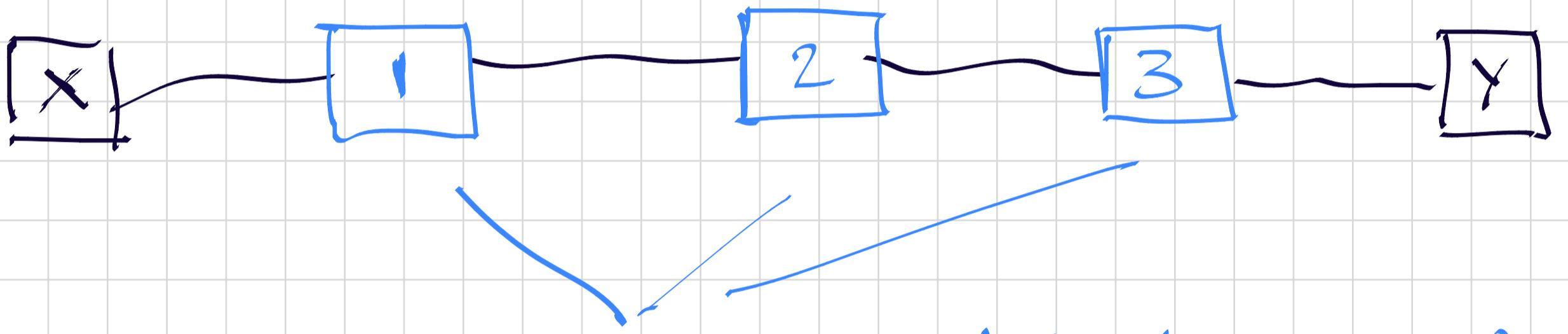
`z = create Spacefactory attrs-f`

type: —,

`attrs-f = { xline: [N,M], yline: [N,M],
xstart: —, xstep: —, ystart: —,
ystep: — }`

some of them
specified →

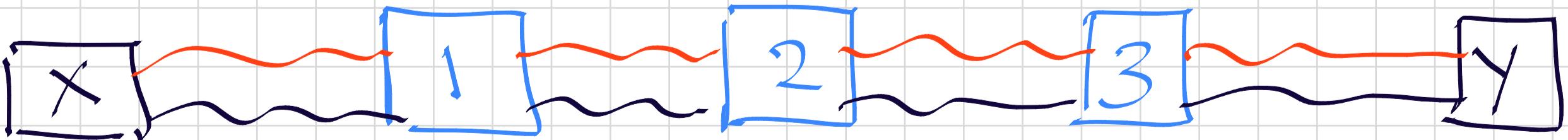
Spacefactory specification forces the space factory to put the spaces at specific locations
→ constrains the waypoints.



generated by the Spacefactory.

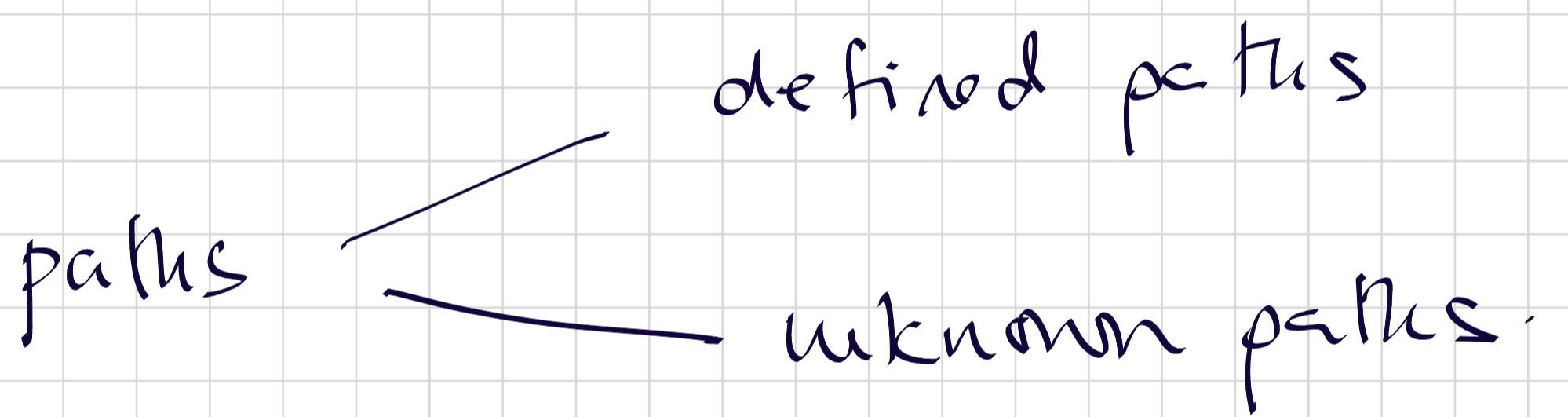
We can also create multiple paths — across the same set of spaces and waypoints.

Two paths connecting the spaces.



Why we want two spaces? Or, why would we want multiple paths?

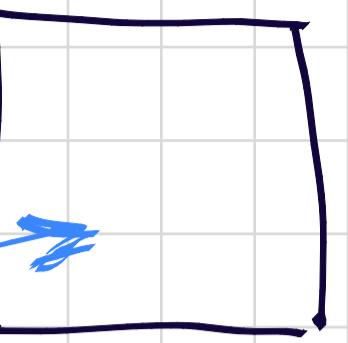
This is to load balance, find minimum hazard paths, or many more reasons... to have multiple paths.



We could compute the interference between a path and other unknown paths.

How can we use this?

Controller



A space is directly created or created using a space factory. Either way we can specify a main function to execute in a space.

Question: What do we call such a main function of a space?

We call it the controller.

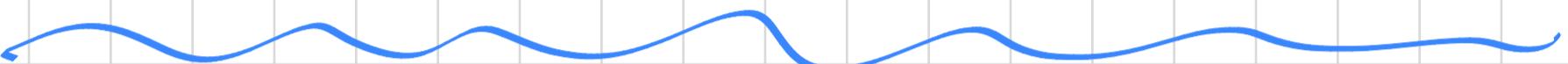
Each space runs a controller.

- * Controller is the main one.
- * We can also have named services).

Key Questions

- What are the key issues to consider in developing a service?
- Each service is a program.
- Space instantiated with a program.

- The service says what to do when an object comes into the space.
 - object has a target? Always a destination?
 - what about object wanting some service?



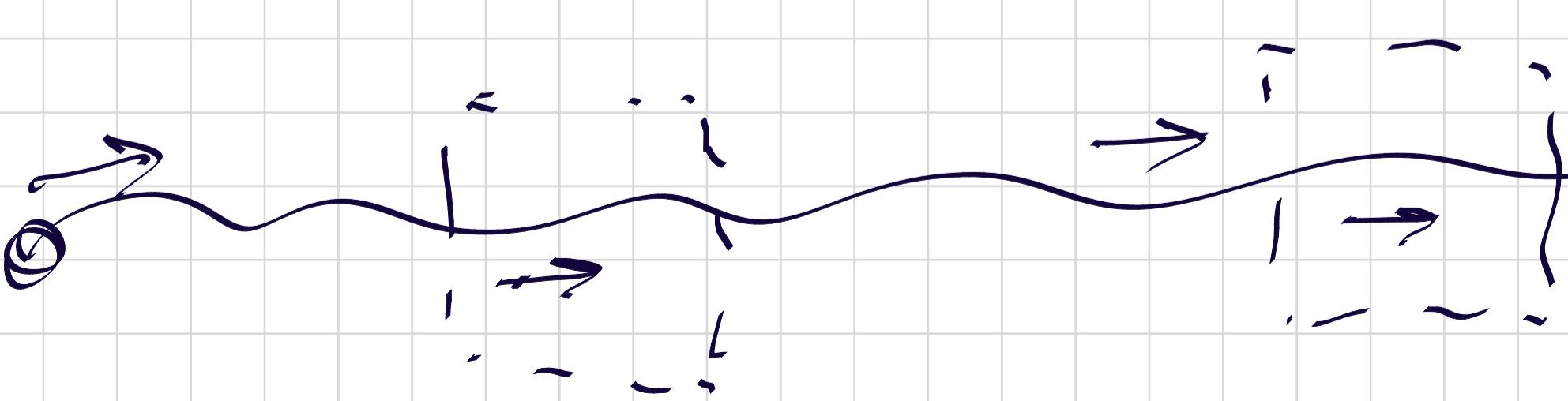
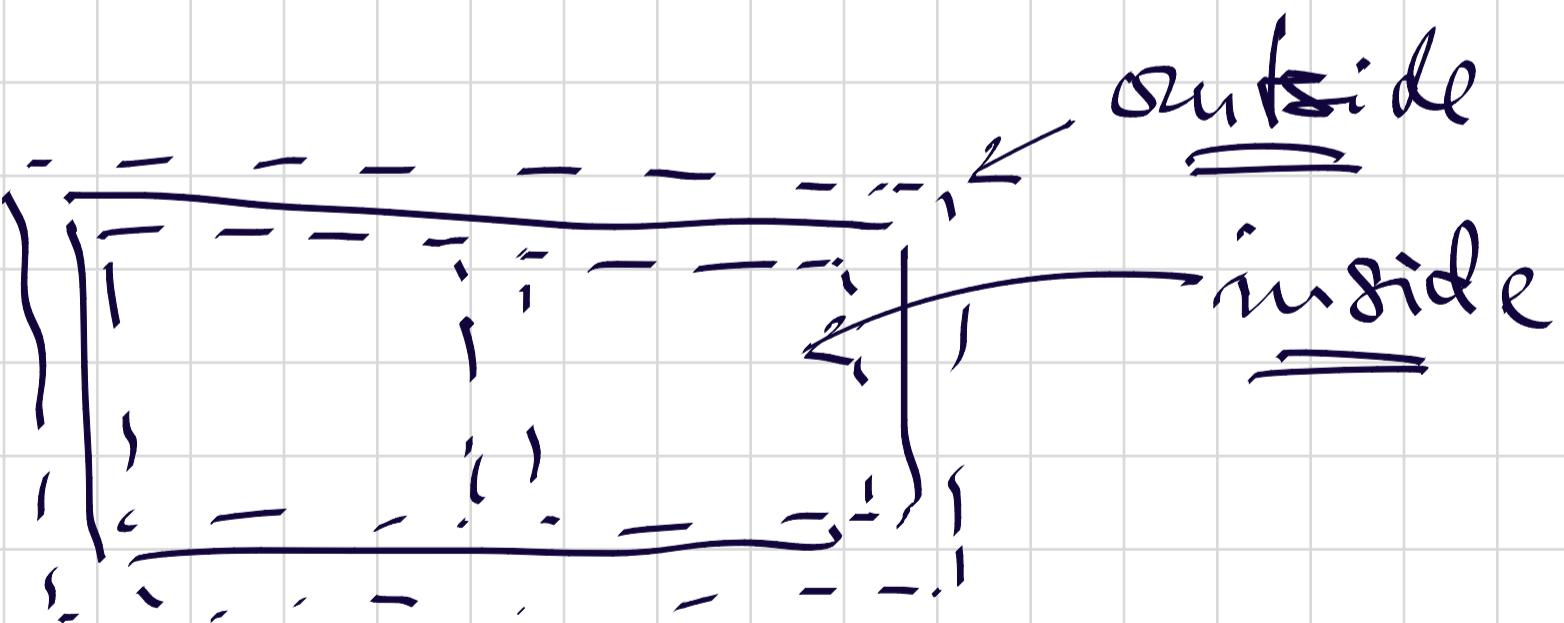
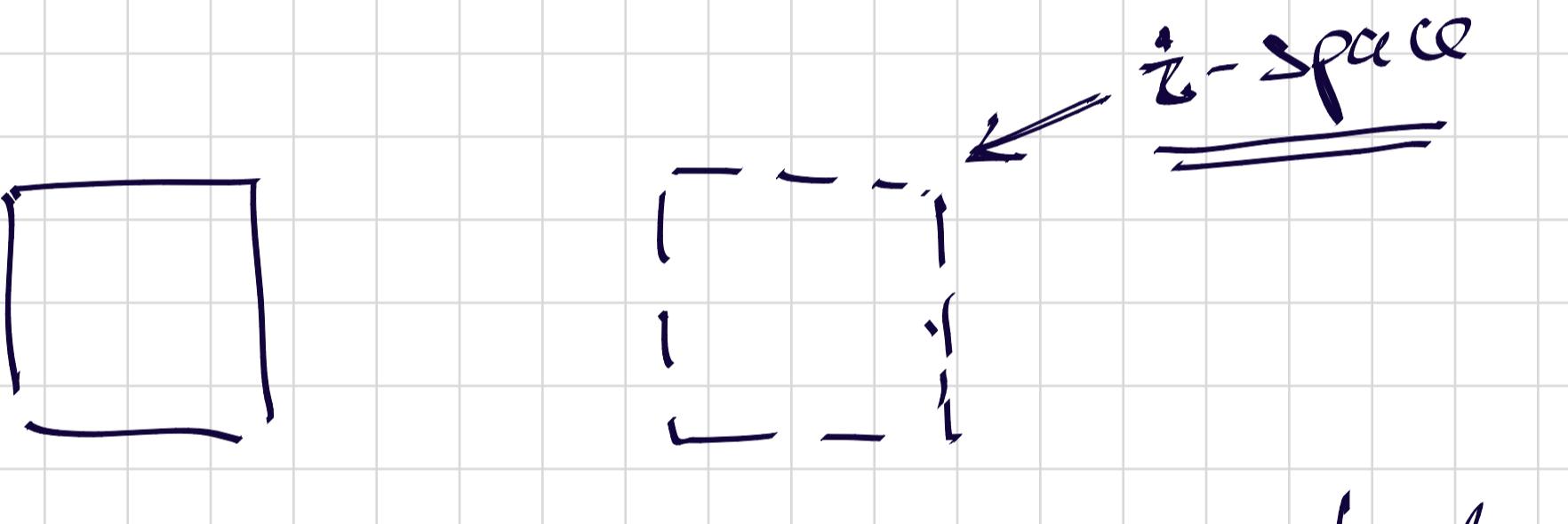
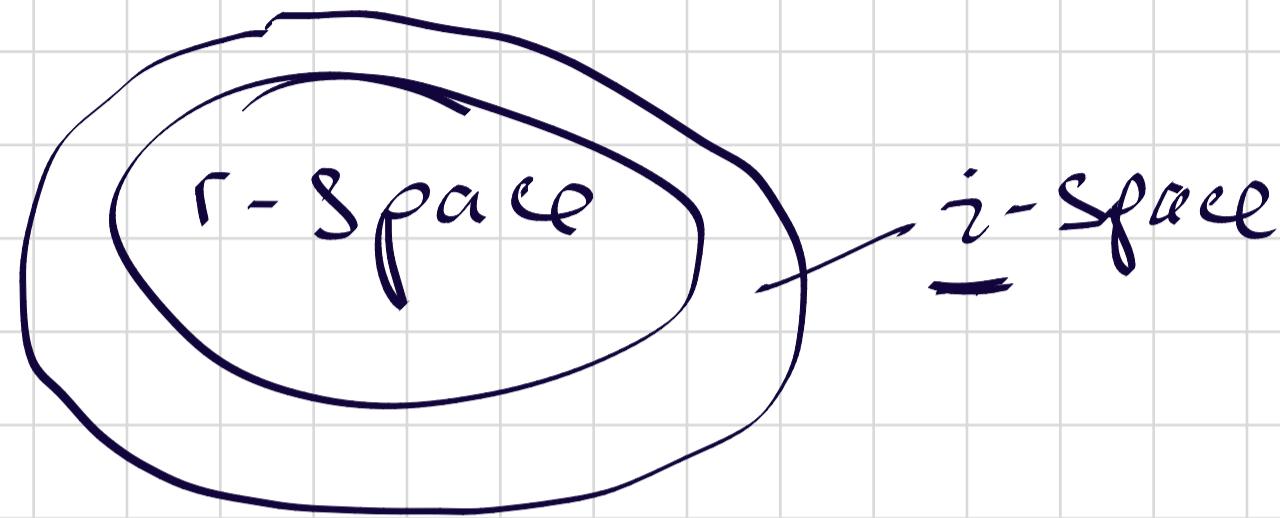
controller vs services (there are many services,
while there is a single controller per space)

Spaces

i-spaces — imaginary spaces

r-spaces — real

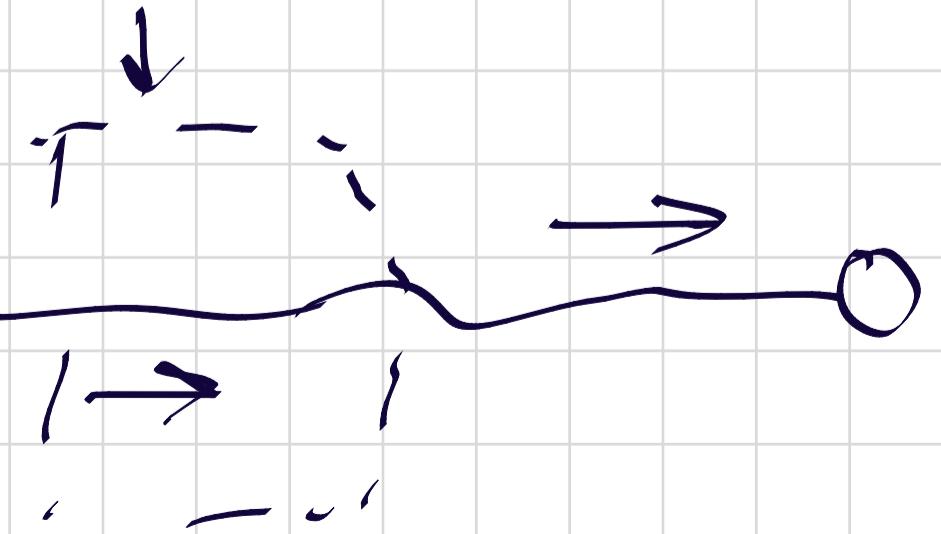
spaces



Can we have
i-space
inside an
r-space ?

Should i-space be
always be outside
or inside an r-space?

server / service
controller

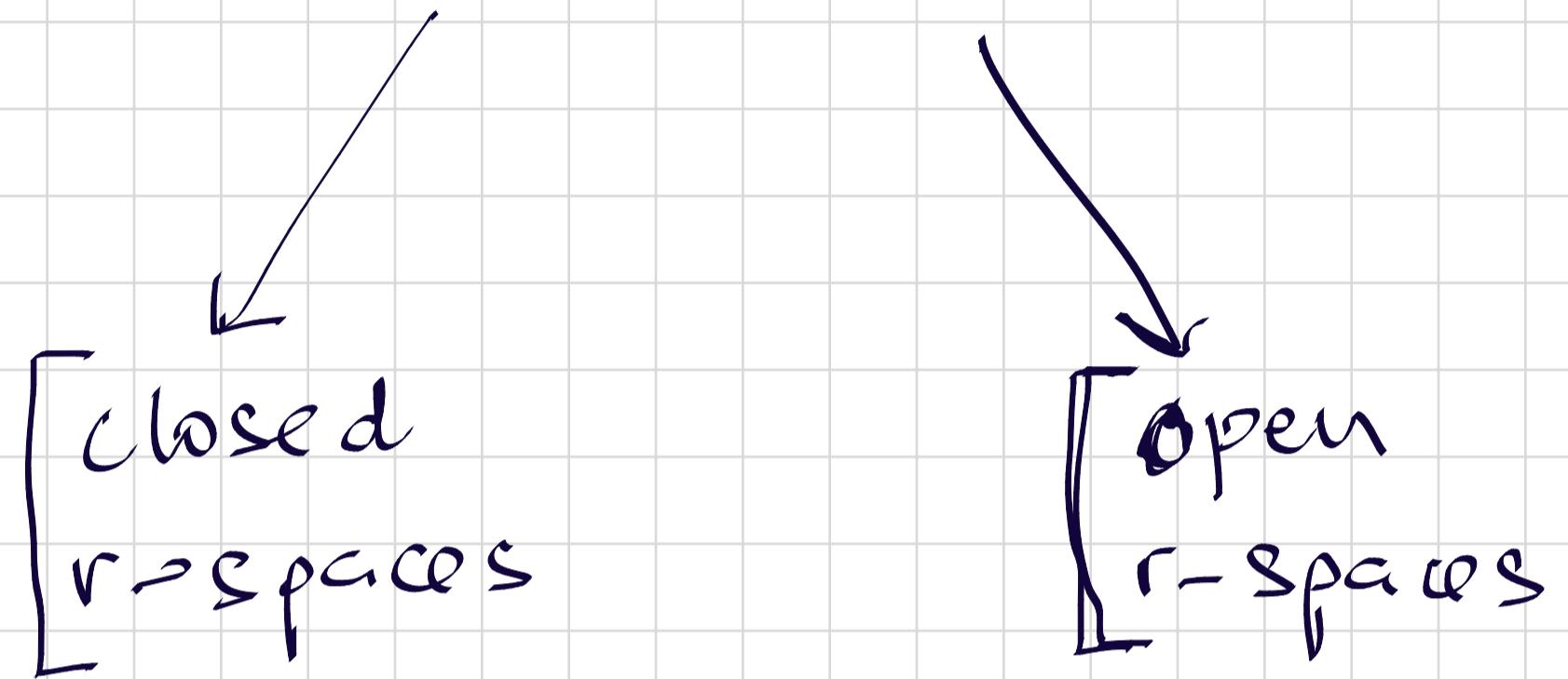


Spaces

① i-spaces - imaginary spaces — these are spaces that do not have a concrete physical realization. So, we designate a portion of a physical space to be an i-space. This is normally done so we can associate a special function with that portion of the physical space. Almost dividing a room to serve different purposes : sleeping space, office space, entertainment (living) space, etc.

② r-spaces - real spaces — these are spaces that have a physical realization that spans the whole space.

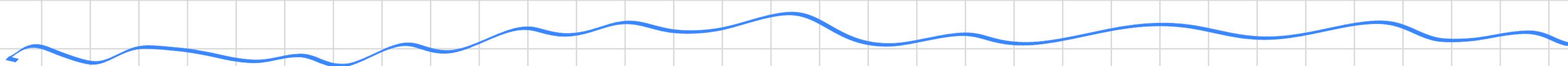
For example, a parking space, storage locker, etc are some examples of a real space.



r-space come with a i-Space with the same dimensions and location - same configuration

r-space } descriptions - in the form of
i-Space } tags, tags describe
the space

how are we going to use the tags?



Input the word facts. — situation of the word

We expect not to change. Invariant situation with respect to the word.

Input the word conditions.

This is the changing part of the word.

facts + conditions = situation



audibiar are generated

