Brown Field Services and Interior Design

DRAFT

What color is your IP Address?

Allan Eising
Network Automation Architect

Telia Company

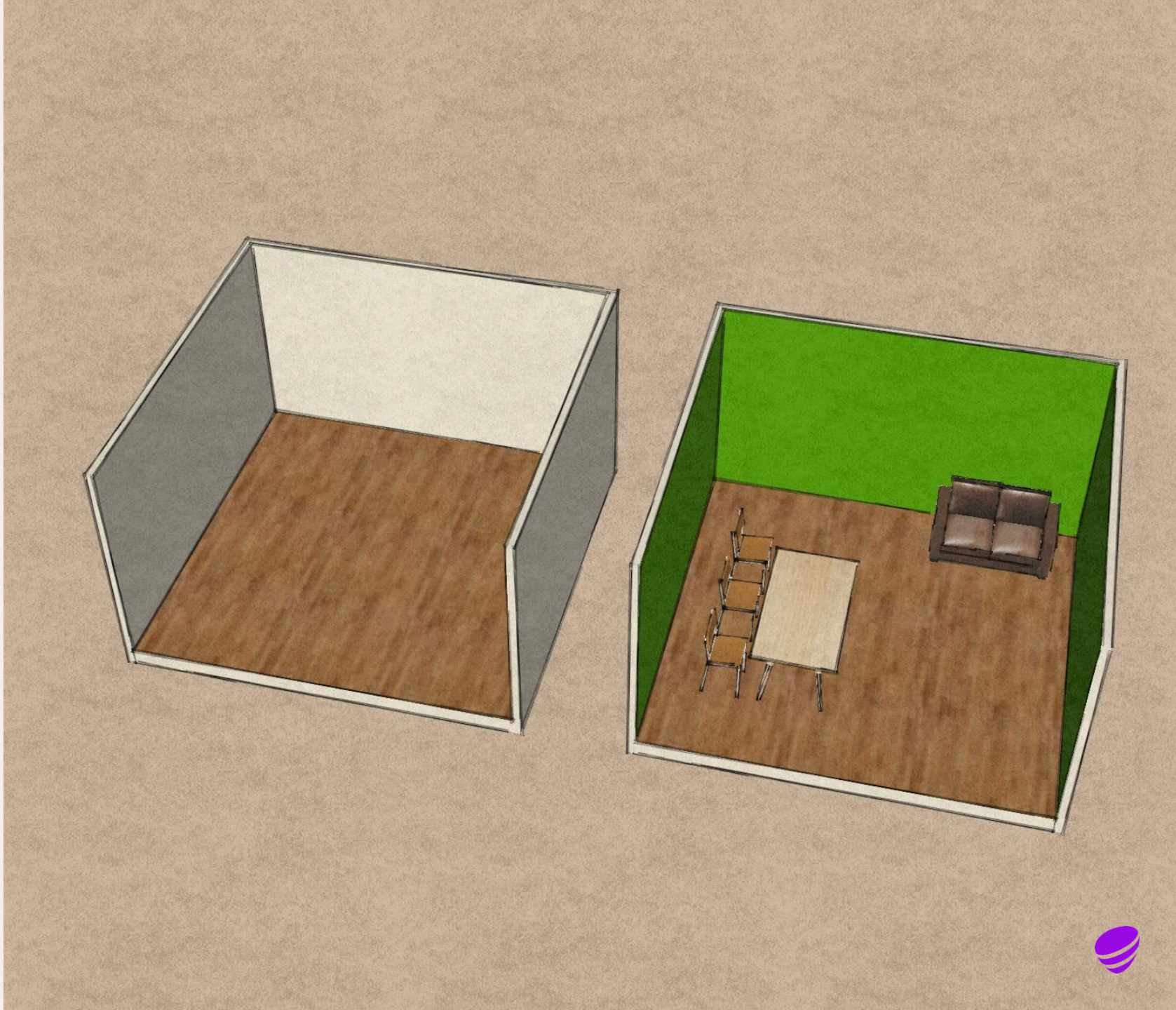
Goal

By the end of this presentation, you should:

— Understand the problem caused by automation in brown-field services

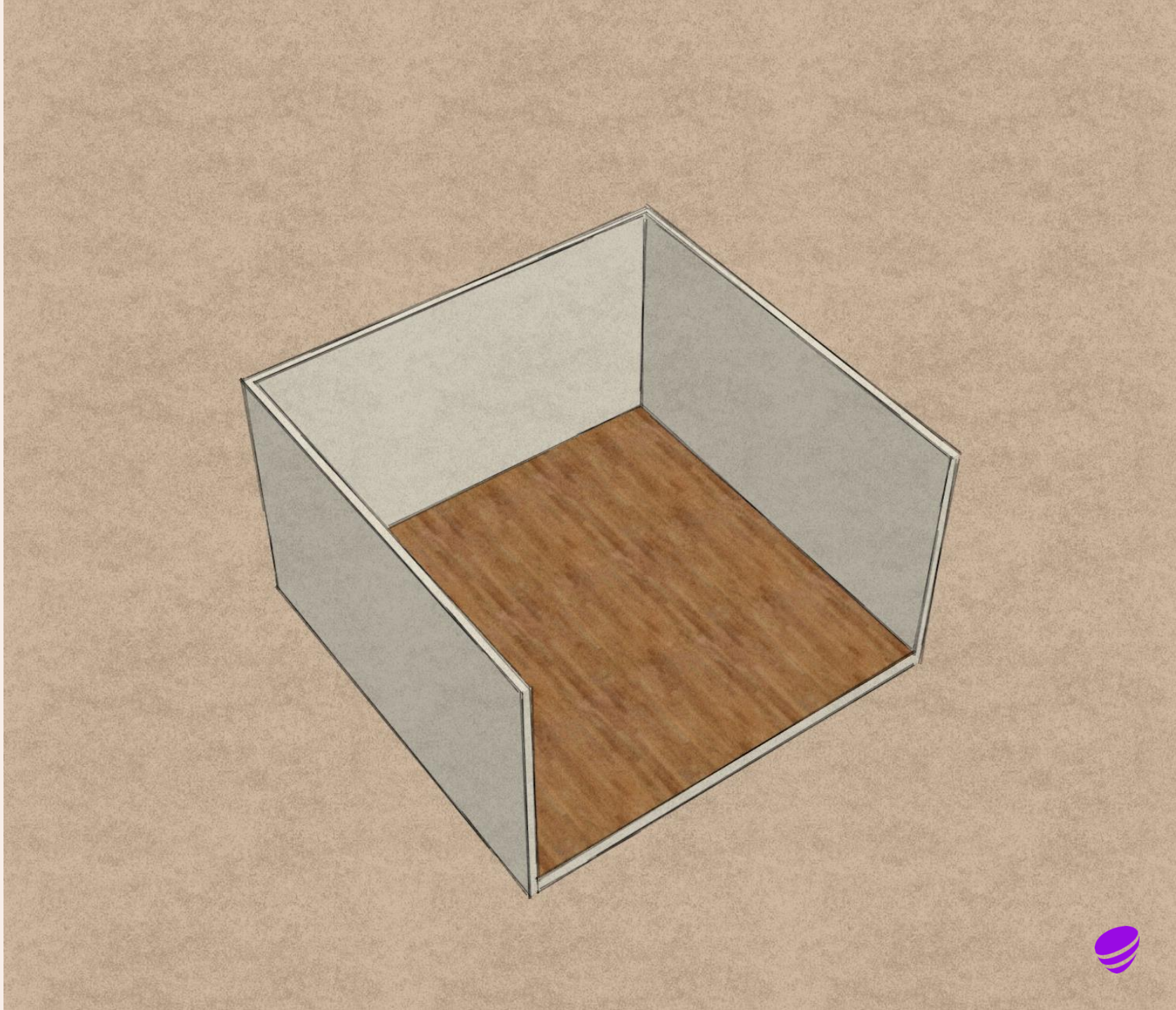
— Understand the process of reconciliation

Explaining reconciliation as interior decoriation



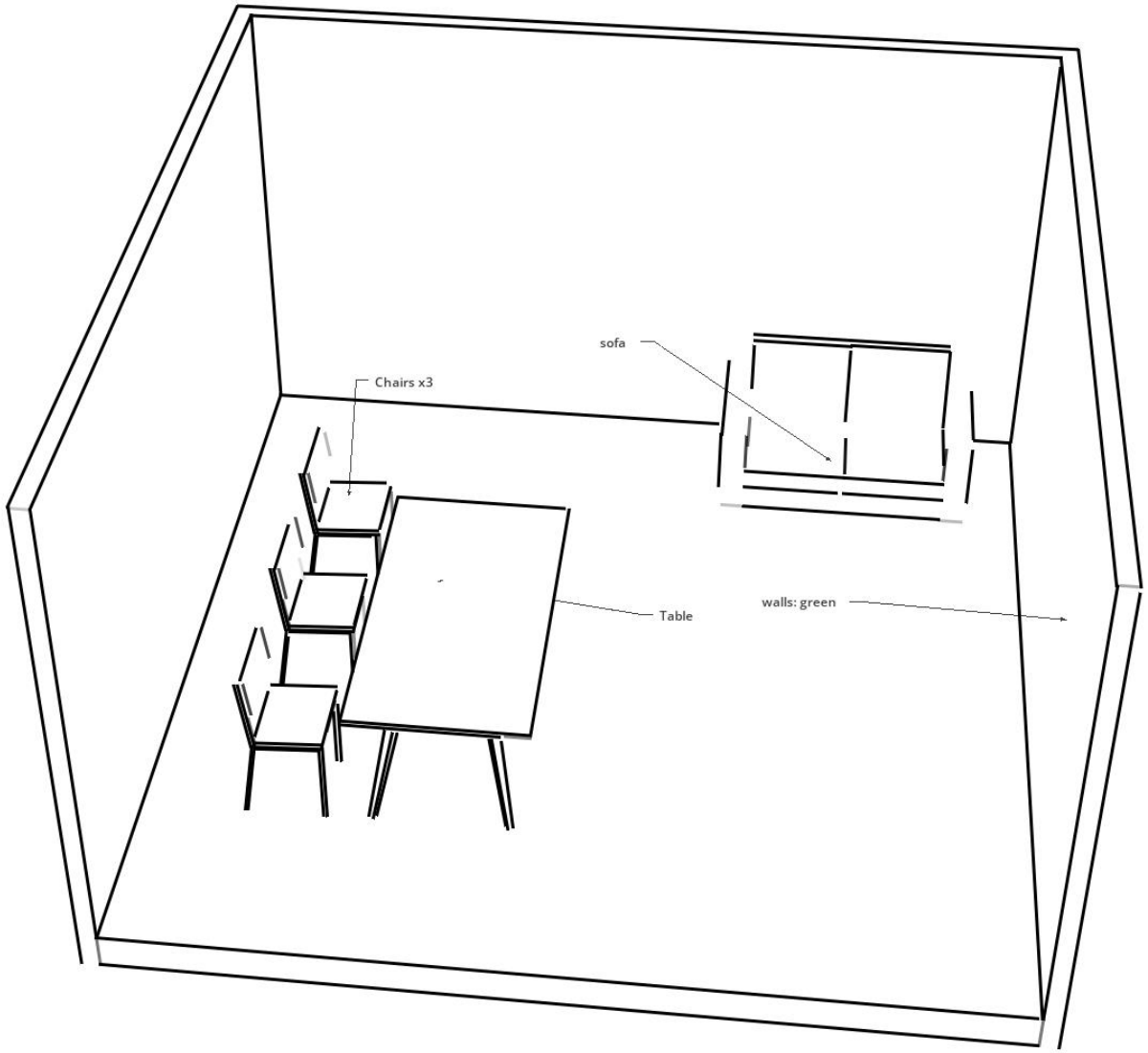
Greenfield

— In a greenfield scenario your automation will start with an empty room
— The walls are white, and no furniture exists



Service

- Services are declarative
- They describe an end result
- It describes how the room should look:
  - Three chairs
  - One table
  - A sofa
  - The walls should be green

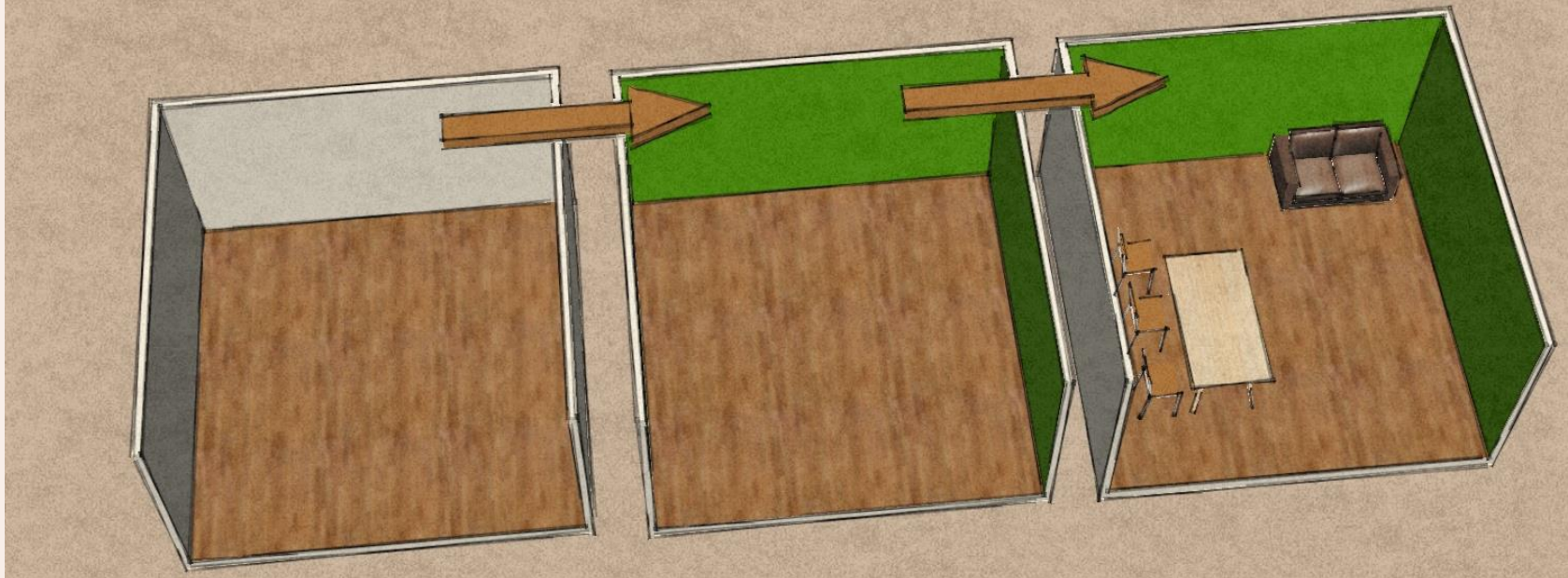


Furnishing

— To get the empty room to the intented state, a number of steps will be taken

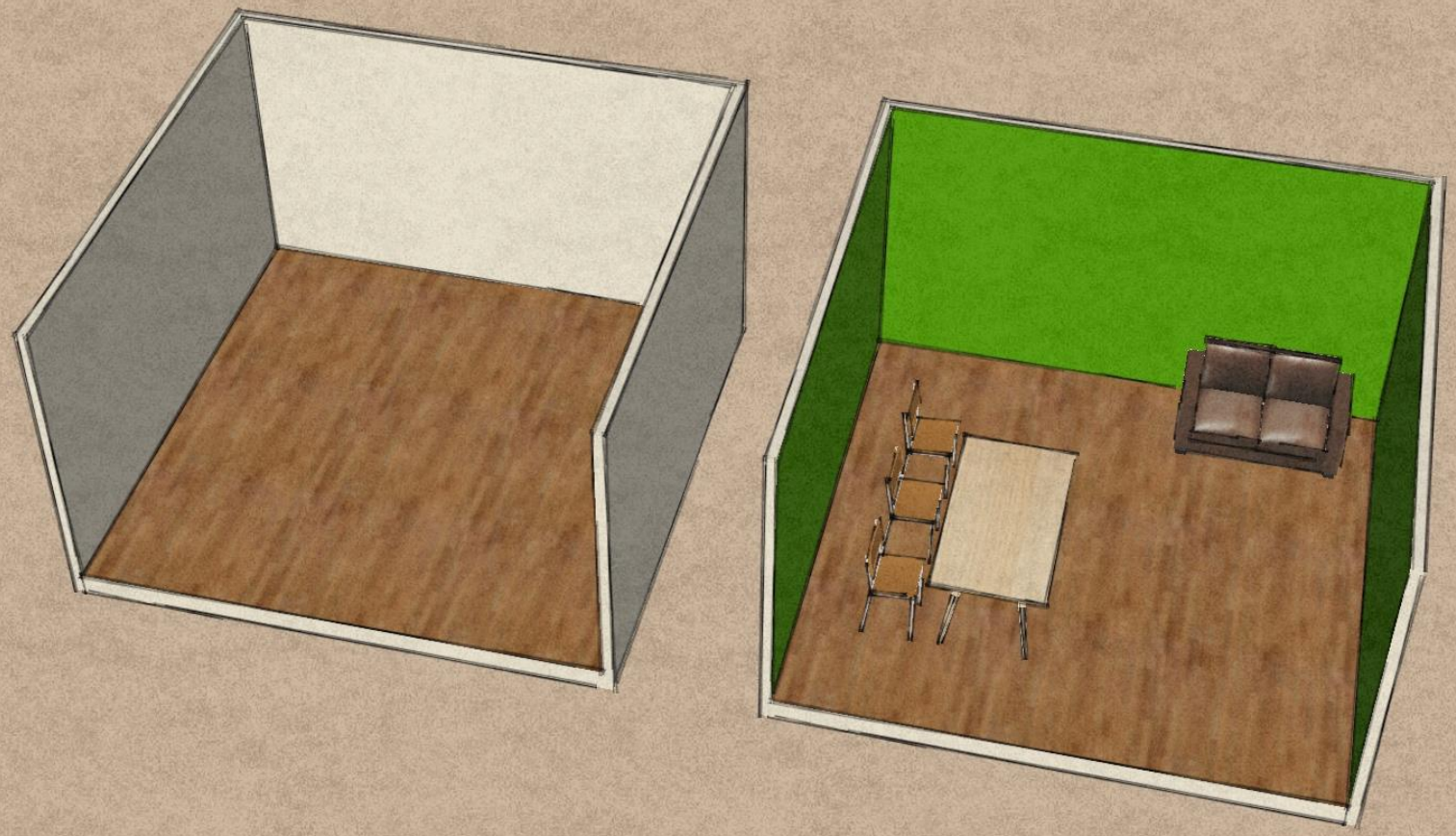
— Here:

1. Paint the walls green
2. Add the furniture



Service Diff

- The service can be recorded as the difference between the two rooms
- This difference can be broken down into a set of steps
  1. Paint the walls
  2. Install three chairs
  3. Set up dining table
  4. Place the sofa



Create, modify, delete

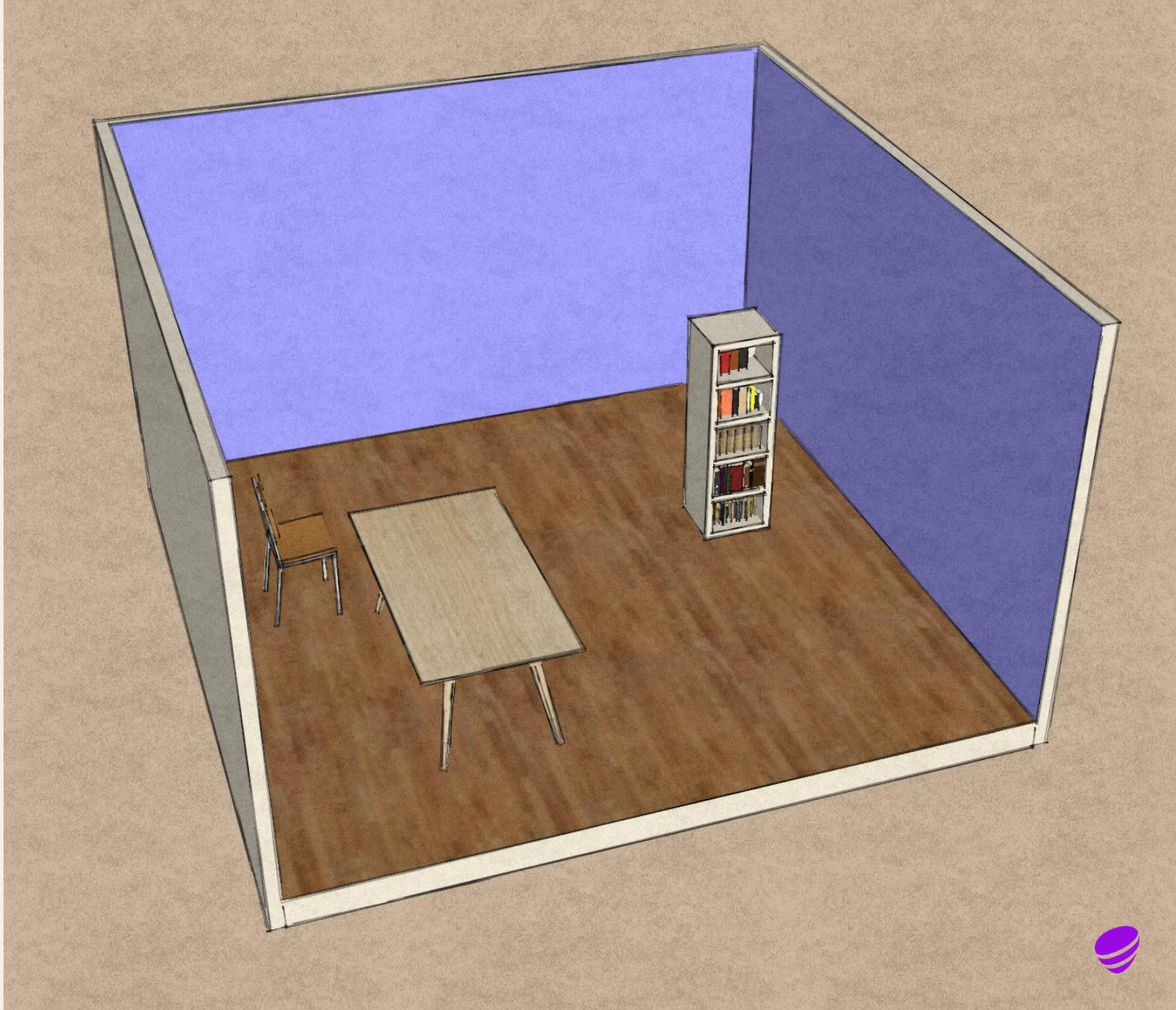
- This diff allows us to have Atomic operations:
- If you record how to go from the empty room to the desired state, you can do the opposite too:
  1. Remove the furniture
  2. Paint the walls back to their original color
- If you modify your service, you just have to remove the old one, and re-apply your modifications

Brown-field

What happens if the room isn’t empty?

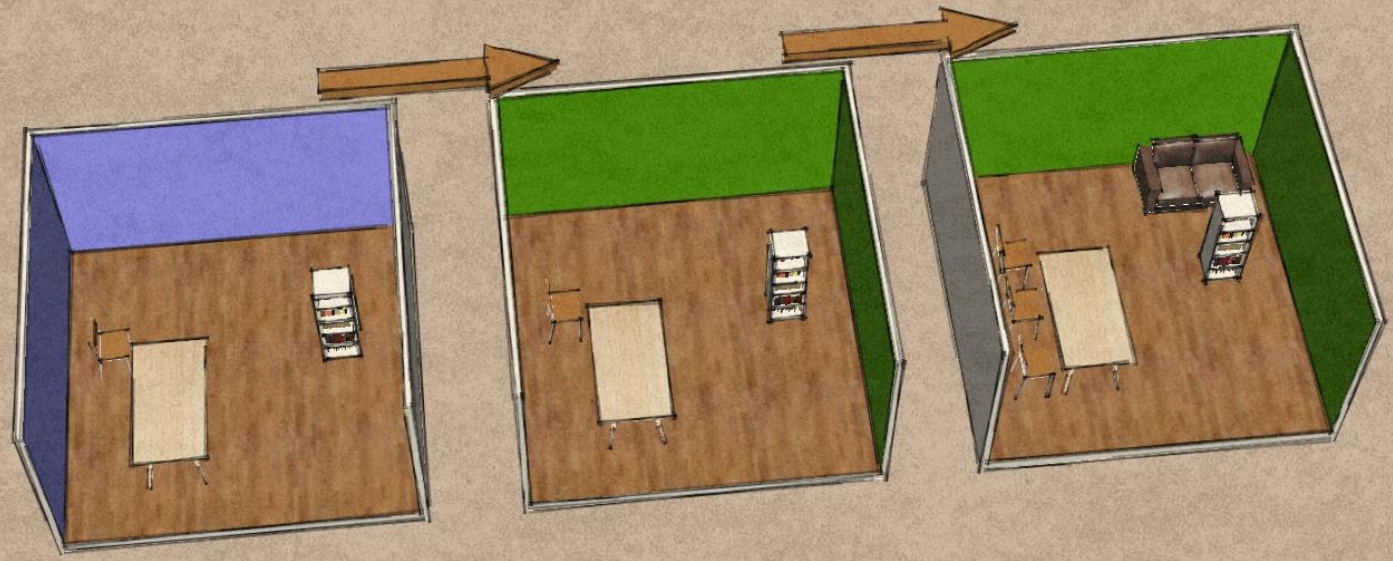
Brown field

— What if instead of an empty room, there is already something in it?
— The walls have a different color
— Some of the furniture is already there
— There’s completely unknown furniture too, a book case!



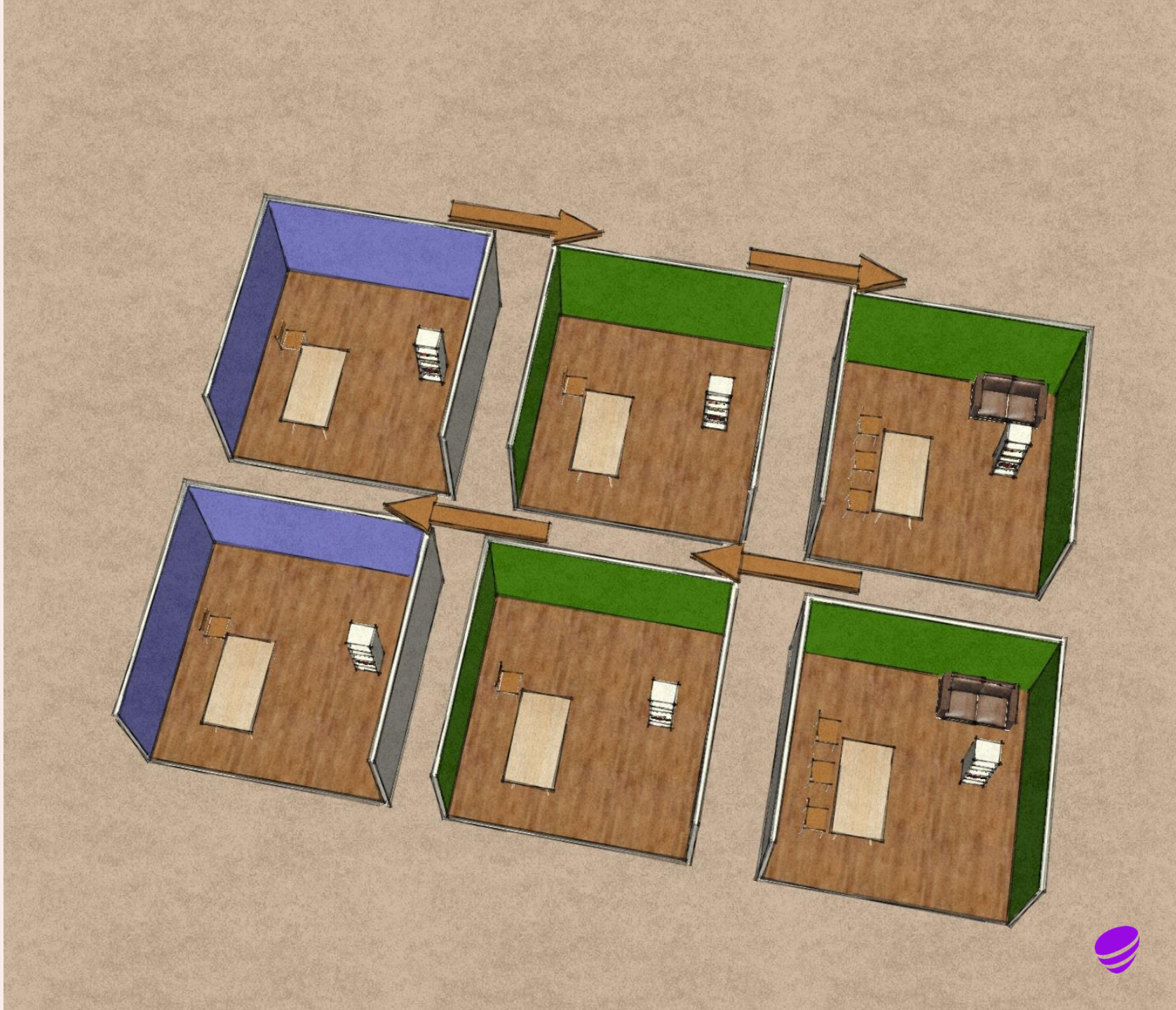
Reaching the intended state

- Your orchestrator should apply the design on top of the original room
- The walls will be re-painted to green
- the missing two chairs will be added
- The sofa will be added
- But the bookcase will remain untouched



Removal

— Removing the service will restore the room to its original state
— Including the old paint and the furniture that was already there



Questions

- The color of the wall represents some configuration that your service includes that had a different value before
- Which color is the right one?
  - The one that was already in the network
  - Or the one that your service applies?
  - Can you name an example of this from the real world?

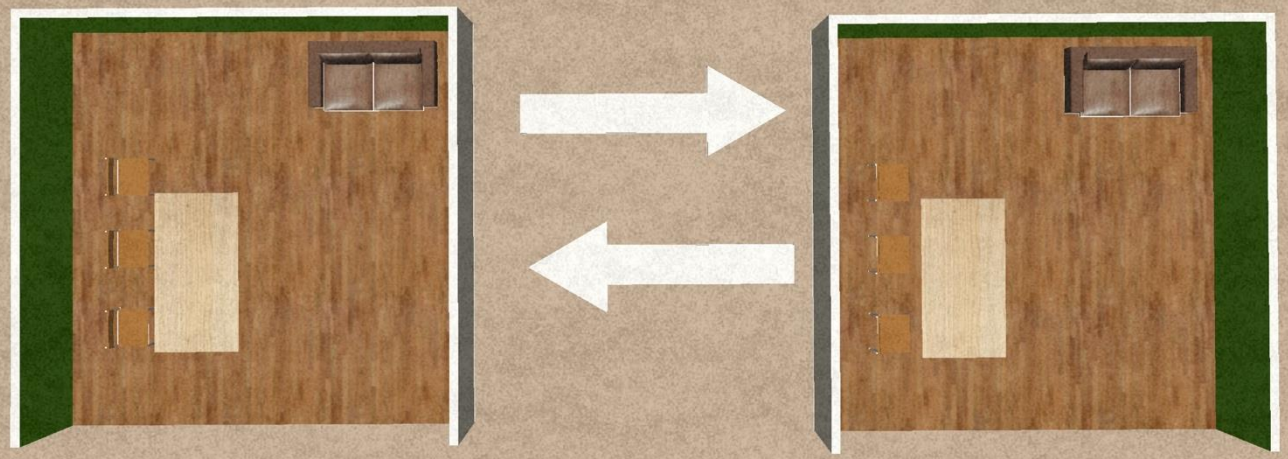
- The bookcase represents configuration that isn’t part of the service
  - Should it actually be there?
  - Should it have been removed by your service?
  - Can you name an example of this from the real world?
  - What would you want your orchestrator to do in your example?

Examples

- Configuration that is different in the network
  - A different VLAN
  - An interface description with the customer name spelled differently
  - A higher or lower MTU value
- Configuration that is not part of the service
  - Something manually configured
  - Something that the router adds automatically

What if it was right from the start?

- When the automation is run, it has actions to run
- But once the service is deleted, there are no steps to reverse
- This means that the service is removed, but the result of the service is still in the network
- We thought we removed the service, but it is actually still there!



Reconciliation

How can we do the right thing?

Reconciliation

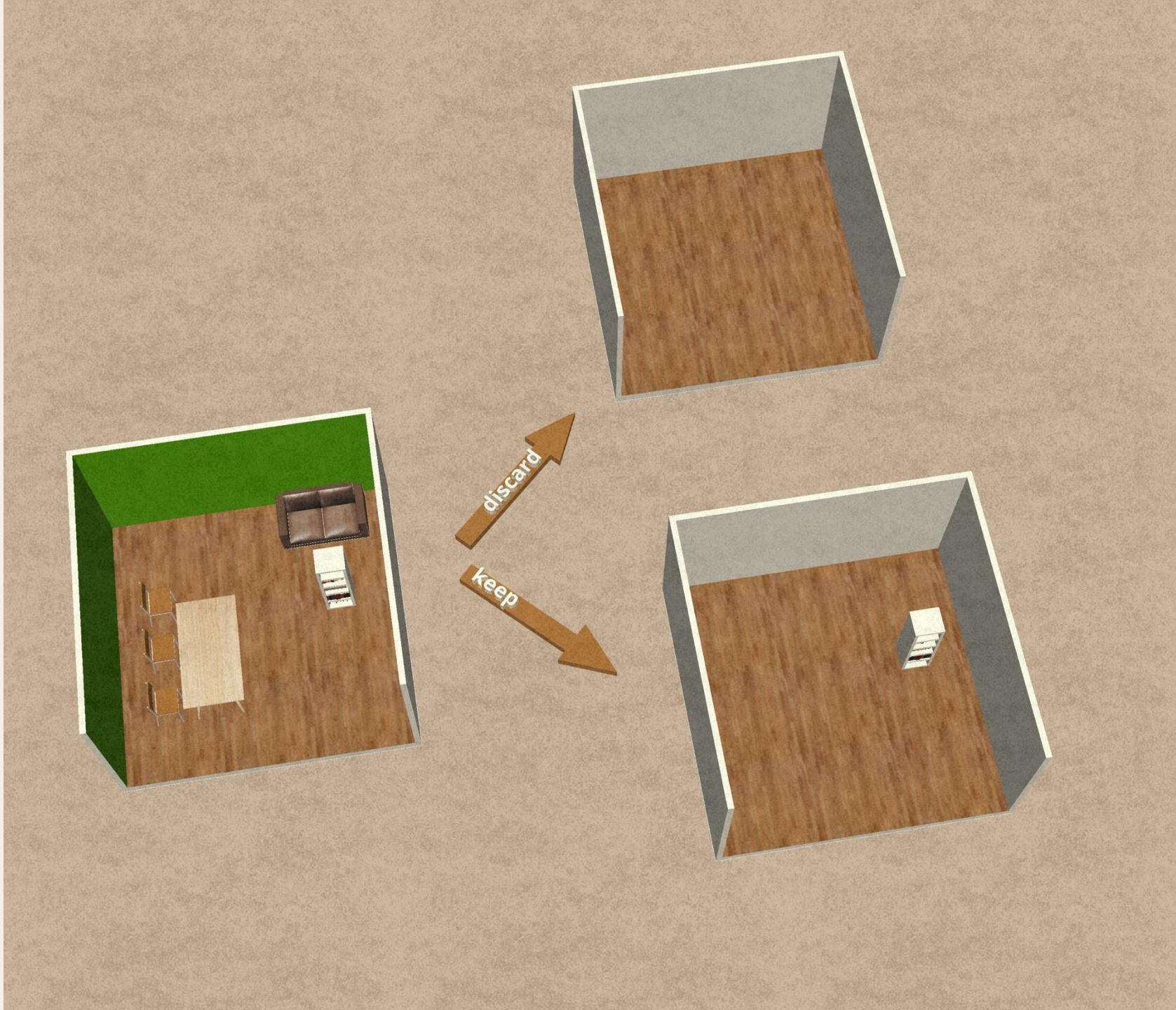
- With cisco NSO, it doesn’t remove what it didn’t create
- So we need to ”trick” it in to believing that it created all of it
- This is done with the “reconciliation” process

What about the bookcase?

— You need to choose what to do with the elements that are not part of your service

— Keep or remove?

— There are pros and cons to both

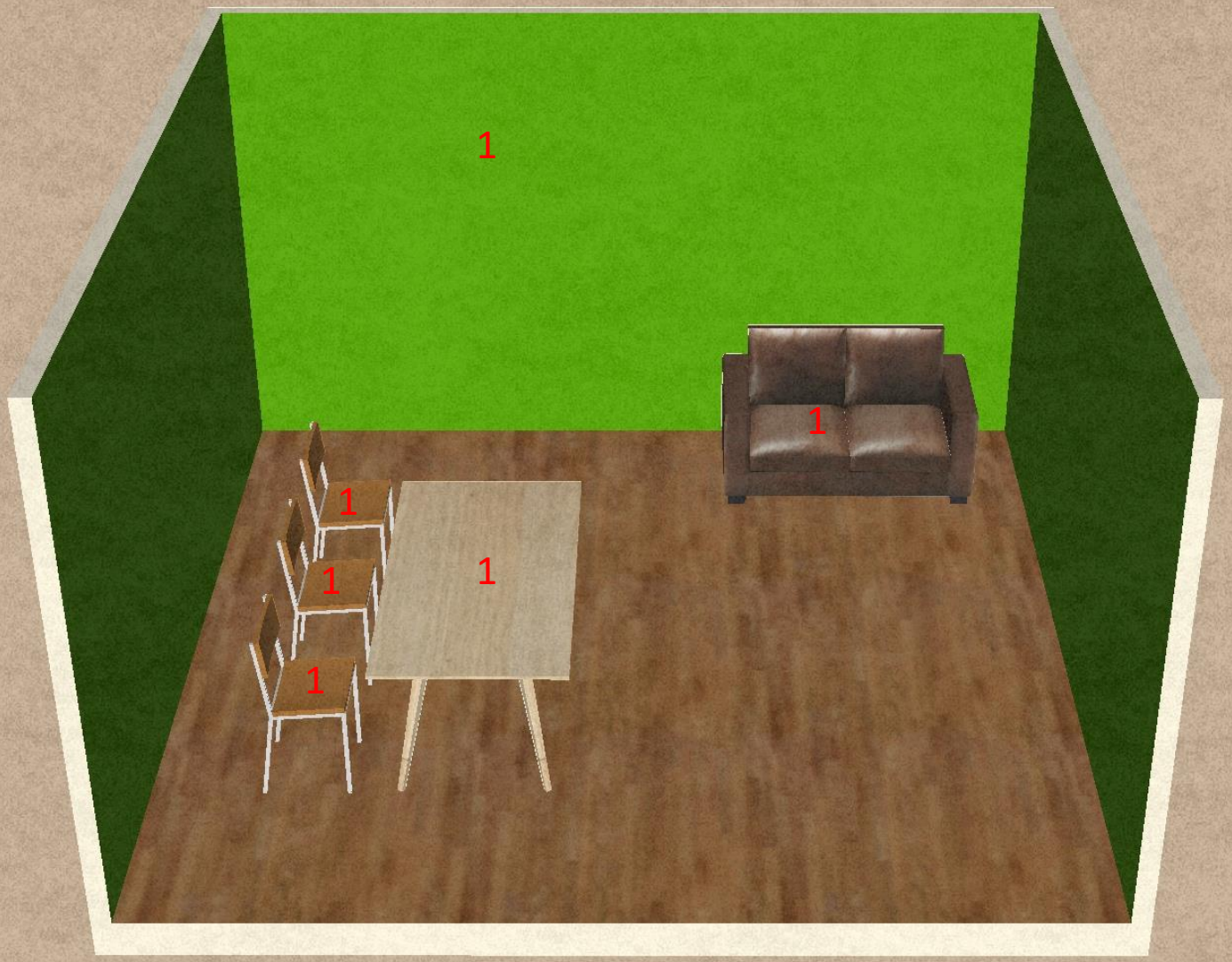


Going a little deeper

Service meta data in NSO

Ref-counts

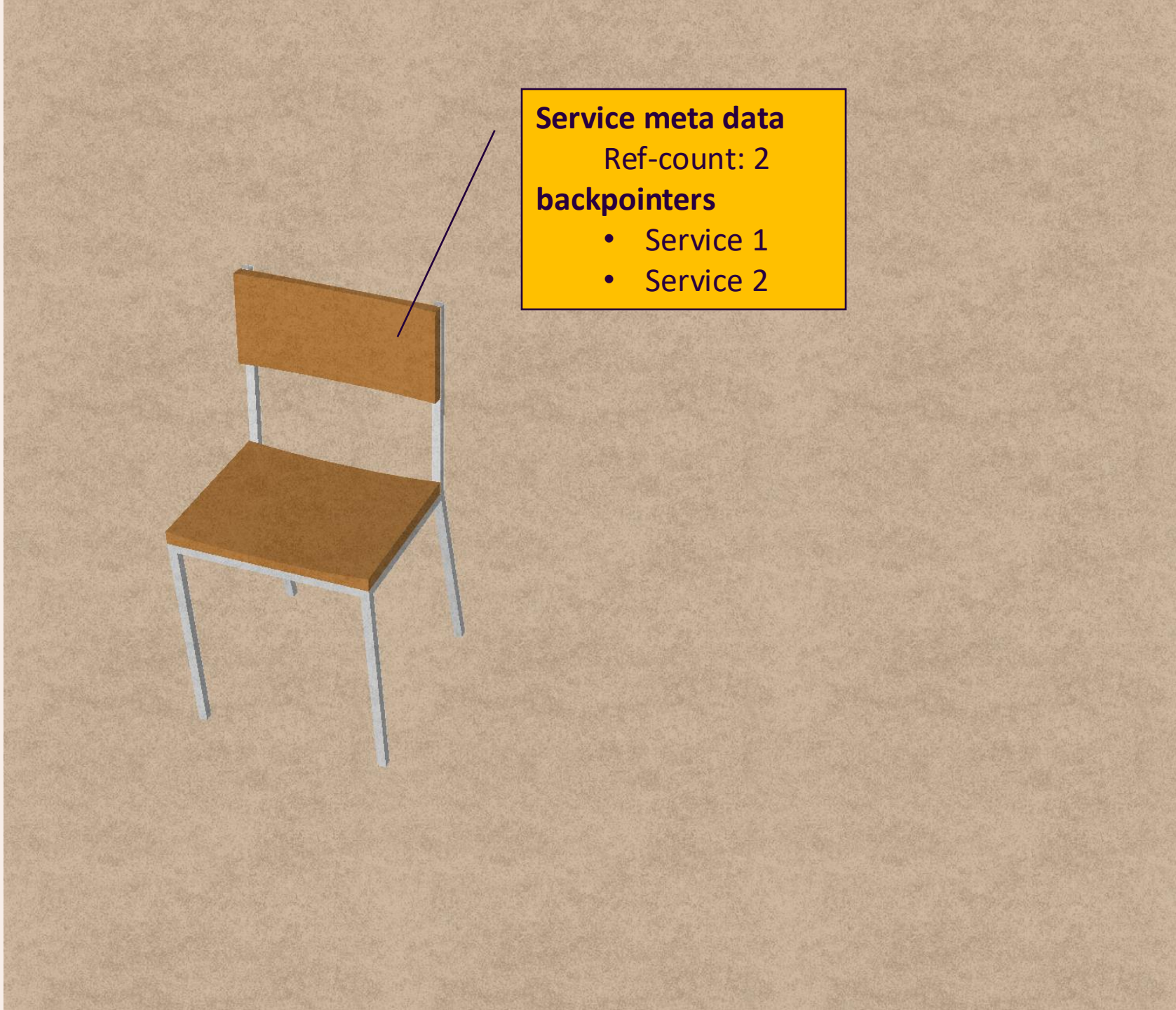
- NSO counts how many times an item is referenced by a service
- If a service adds an item that doesn't exist, it has count **1**
- If a service wants to add something that is already there, it adds **1** to the ref-count
- If a service is removed, and the counter reaches 0, the item is removed



Backpointers

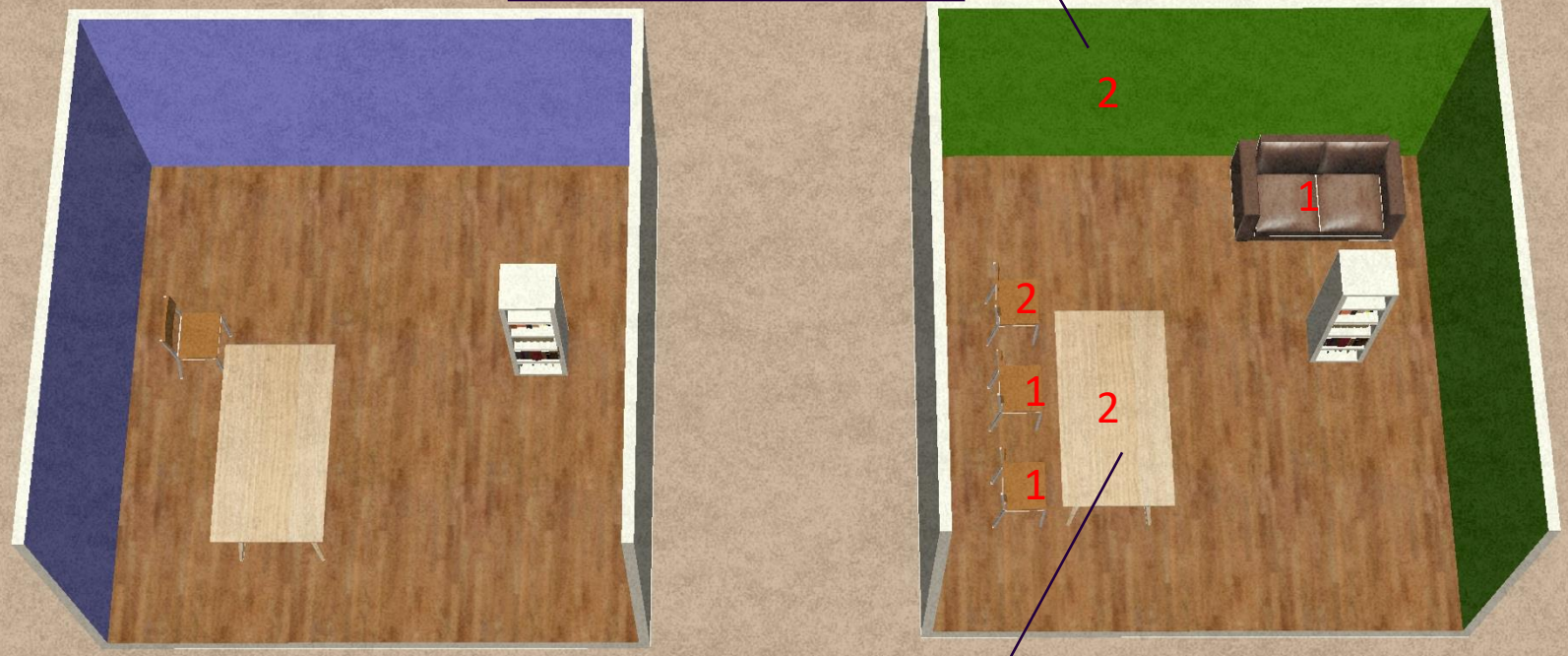
— Each service also leaves its address on each item

— This allows us to tie the ref-count to the actual services



Brownfield

- If something already exists, it will get a ref-count but no backpointer
- The bookshelf is not part of the service, so it will have no ref-count
- Original values of items changed will also be recorded



Well actually…

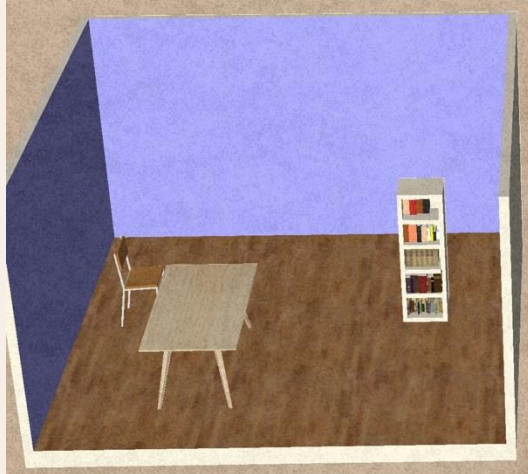
— It is possible to “hide” some of the changes from NSO, for example:

— We can create things that are not removed afterwards
— We can do a “no shutdown” on an interface without making NSO issue a “shutdown” on removal.

Resolving differences – why is this hard?

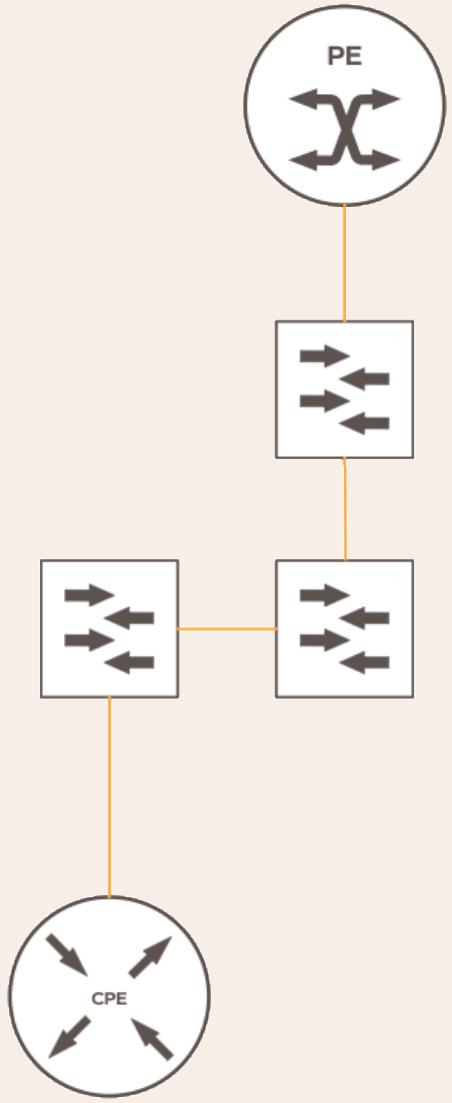
Who is right?

- Was the wall actually supposed to be blue?
- How many chairs did the customer actually want?
- Did the customer actually order a book case but we didn’t support that in NSO or our standard product?
- This requires some insight in to the customer order



Who is right? (with proper examples)

- Was the ~~wall~~ vlan actually supposed to be ~~blue~~ 101?
- How many ~~chairs~~ megabit/s did the customer actually want?
- Did the customer actually order ~~a book case~~ OSPF but we didn't support that in NSO or our standard product?



Challenge at scale

— Some of the original states might have been correct
— Some are obviously incorrect
— But it may be hard to tell without knowing
— There may be thousands of services to analyze

Summary

- Recording the difference in state is a very powerful way to run an orchestrator, as it allows:
  - Using a declarative model that works together with what is already in the network
  - Recording steps taken to create the service, and reversing it when deleting
- However
  - This can lead to partial or complete configurations being left when deleting services
  - Misconfiguration might hide in the state before the service was created
  - Configuration not handled by the service could be left behind
- You must decide what to do:
  - Unhandled config can be kept or removed.
  - Values that were different should be understood, as they will disappear after reconciliation
  - This can be a challenge at scale.