Embedded Workspaces in PMR2

Tommy Yu
Auckland Bioengineering Institute, University of Auckland

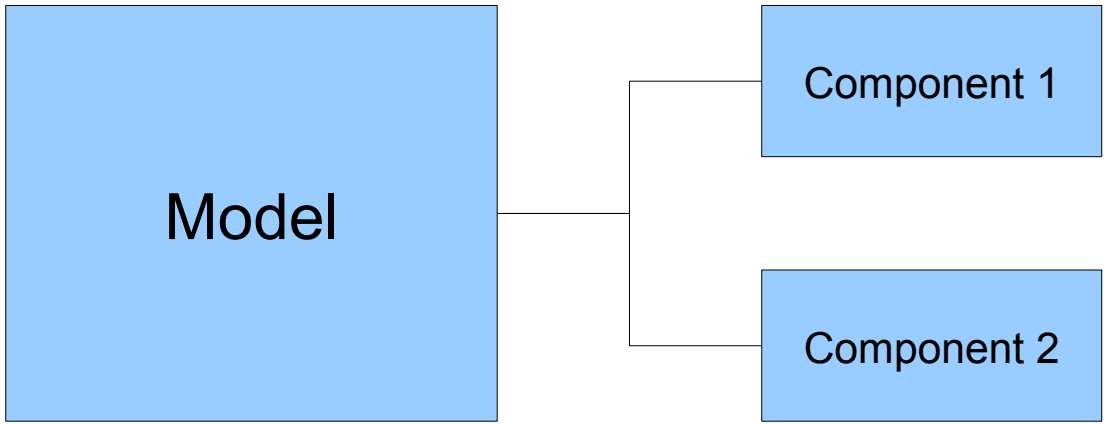
AUCKLAND BIOENGINEERING INSTITUTE
THE UNIVERSITY OF AUCKLAND
NEW ZEALAND
Te Whare Wānanga o Tāmaki Makaurau

Introduction

- PMR2 introduced this concept of workspace for storage and sharing of models.
- This enabled the storage of CellML 1.1 models as the previous repository (PMR1) did not support this.
- The most recent PMR2 release (v0.2) introduced various usability improvements for the modellers, and added in support for embedded workspaces.

What are embedded workspaces?

- They are basically a workspace inside a workspace.
- In a way it is similar to model encapsulation.
- Not really that much different than a standard workspace, aside from the fact that they are referenced by another.



Why embedded workspaces?

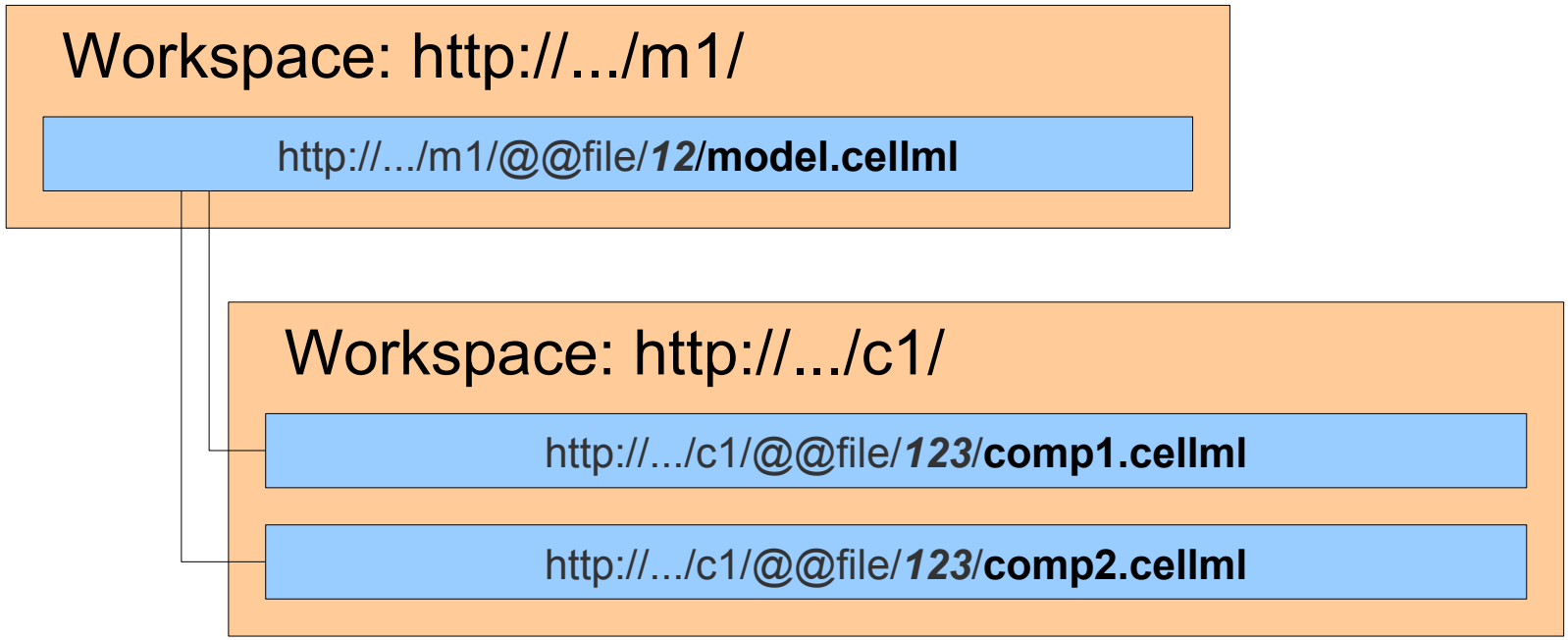
- Manages the separation of core model from its subcomponents.
- Allows sharing and reuse of model components separate from its source models.
- Version pinning of imported components.
- Makes building a shared library of components possible.
- Import of components via relative URI.

Reuse with embedded workspaces

- A model can have its shared component split up.
- The components would go into a new workspace.
  - This can be thought of as a kind of encapsulation for making the component easier to use by others.
- The original workspace will then embed this new workspace.
- May require changes to the import link within the top level model file depending on new organization.
  - Should not be an issue if components were already separated out in some kind of directory structure.

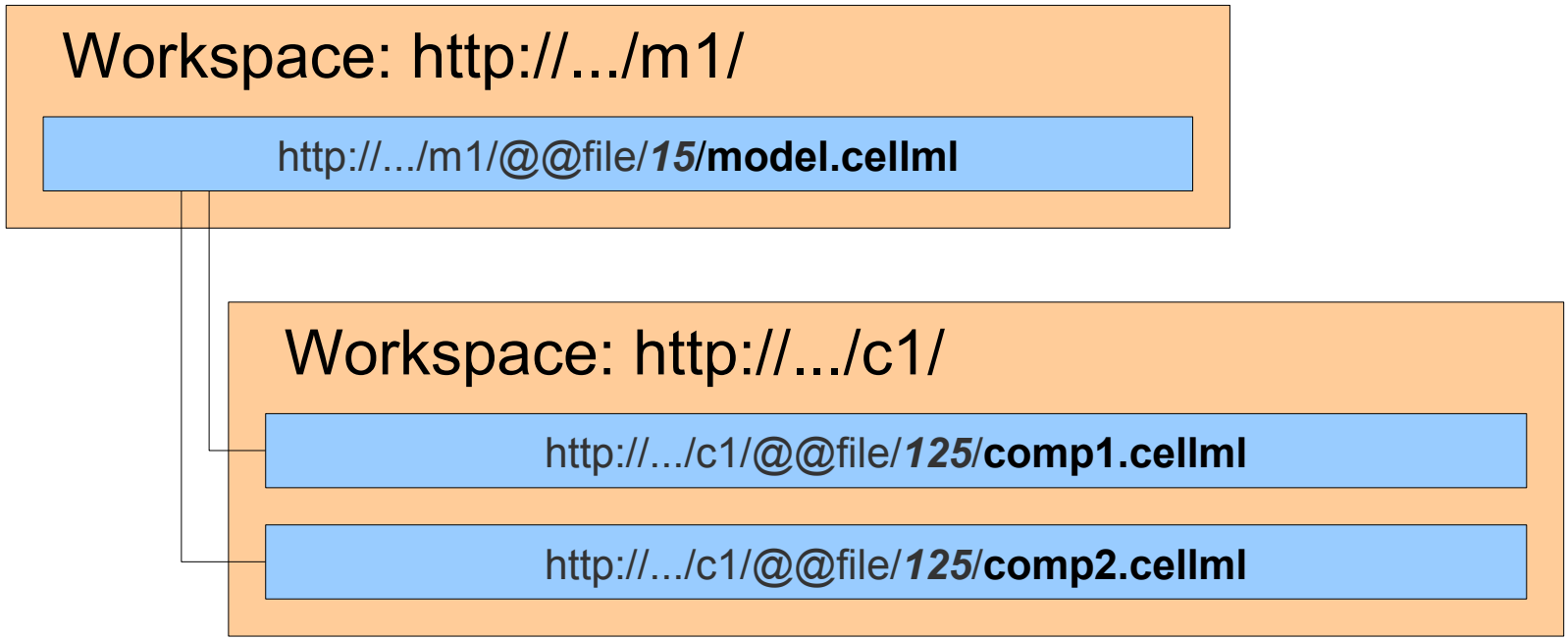
Importing other workspaces

- We have a workspace 'm1' with a model file, that needs to import components from 'c1'. Without embedded workspaces, absolute links must be used.



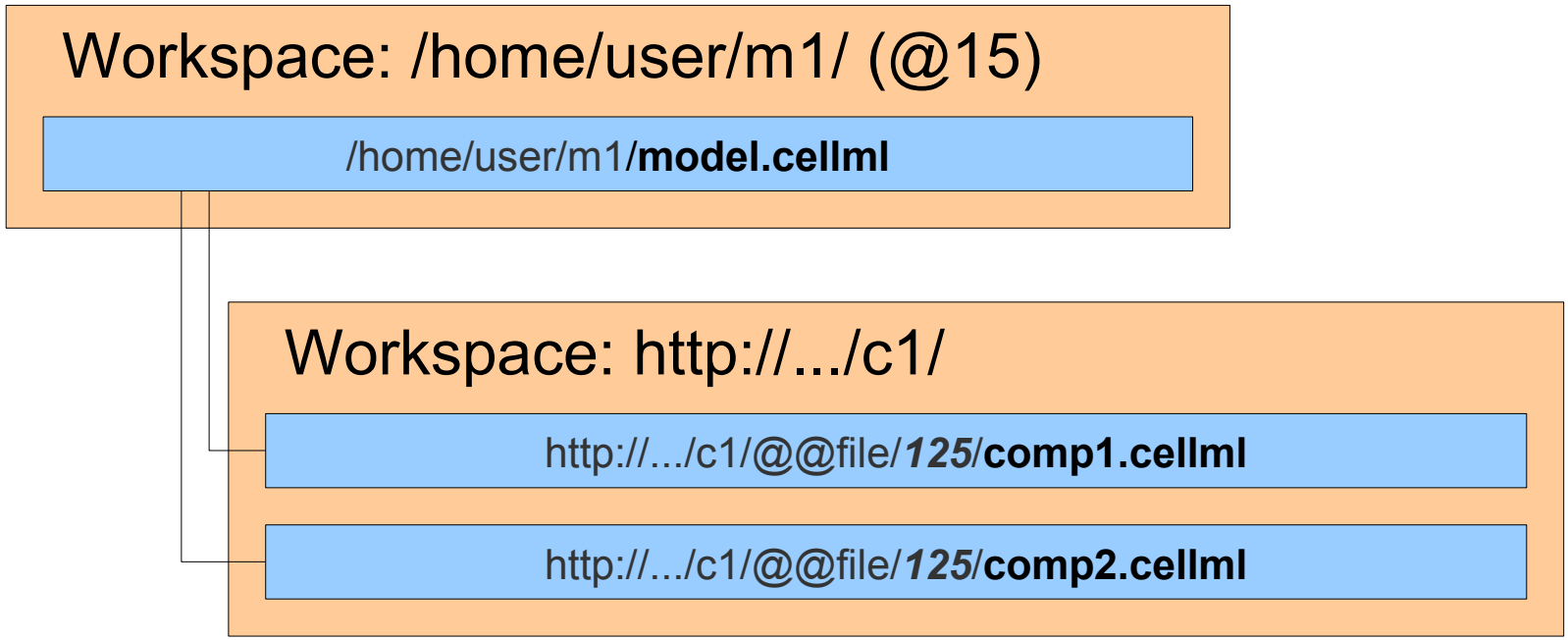
Importing other workspaces

- Absolute import links are cumbersome, modellers have to update each link manually if they need to import a newer (or other) versions, every time.



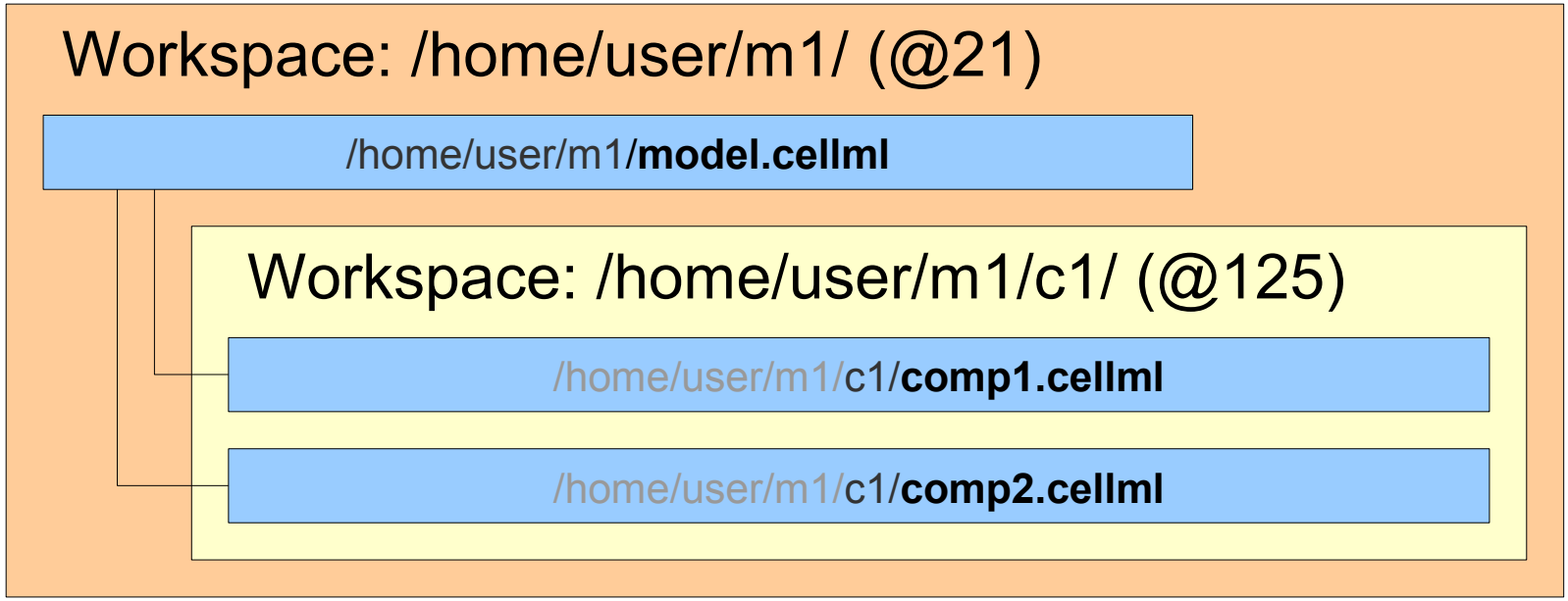
Importing other workspaces

- Also this means a local copy of the model will require a constant connection to the live repository for it to work.
- Cannot easily edit components.



Importing other workspaces

- Using embedded workspaces will enable imports via relative URIs, a significantly more portable solution.
- On the local filesystem, they exist as a hierarchy of directories.

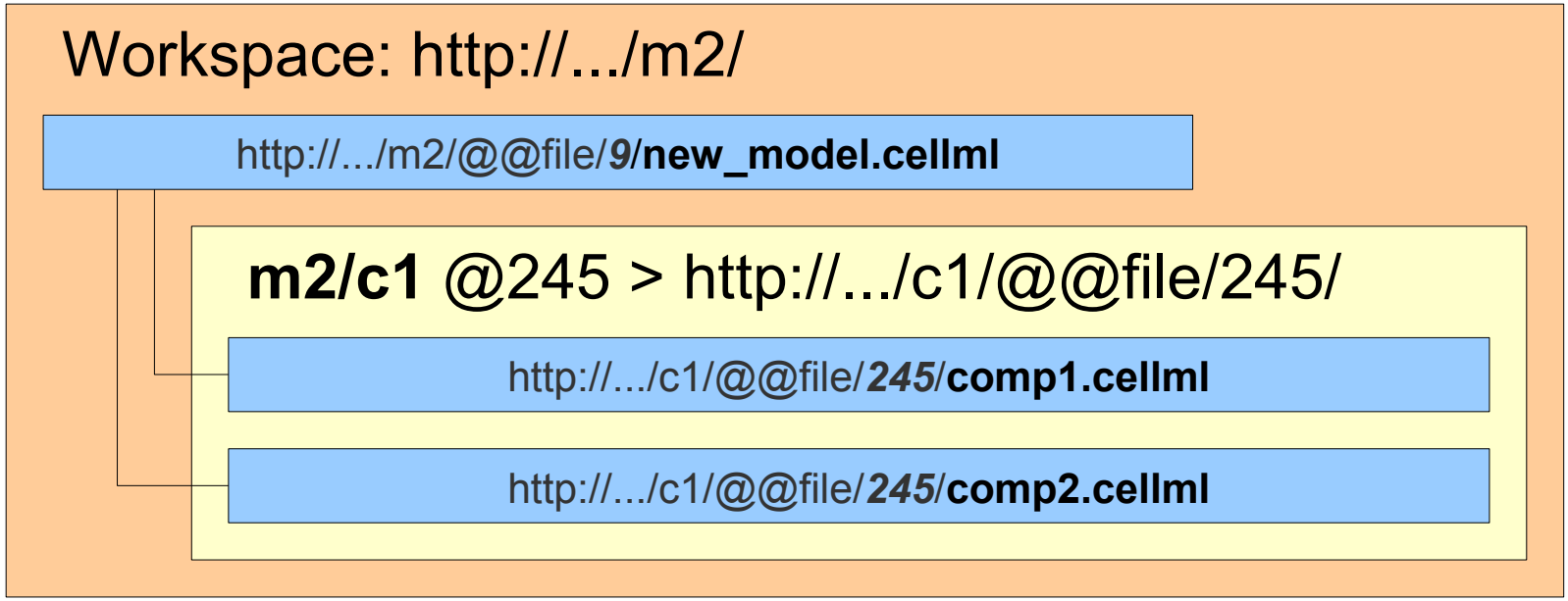


Importing other workspaces

- Once pushed back to the repository, the relative links will have new references to its source workspaces
- PMR2 will redirect the client to the correct revision of the file when it attempts to import using the relative links.

Sharing components

- Another modeller may reuse the components that was split up previously.
- They will follow on the same process described in the previous slide to include the models.



Version pinning

- The author of the components may make changes to shared library.
- This may introduce incompatibility with other existing models that depend on it.
- Pinning an imported component to a specific version removes this surprise.
  - So no modeller can yank the carpets that other modellers might be standing on.

Shared library of components

- Once all the models have their components split out from its core into separate workspaces, we can then group them together.
- There will be exposures of each of these components to introduce their functions.
- Listings out of them will be generated, forming the library of shared components.

Fin

- Thank you for your attention.