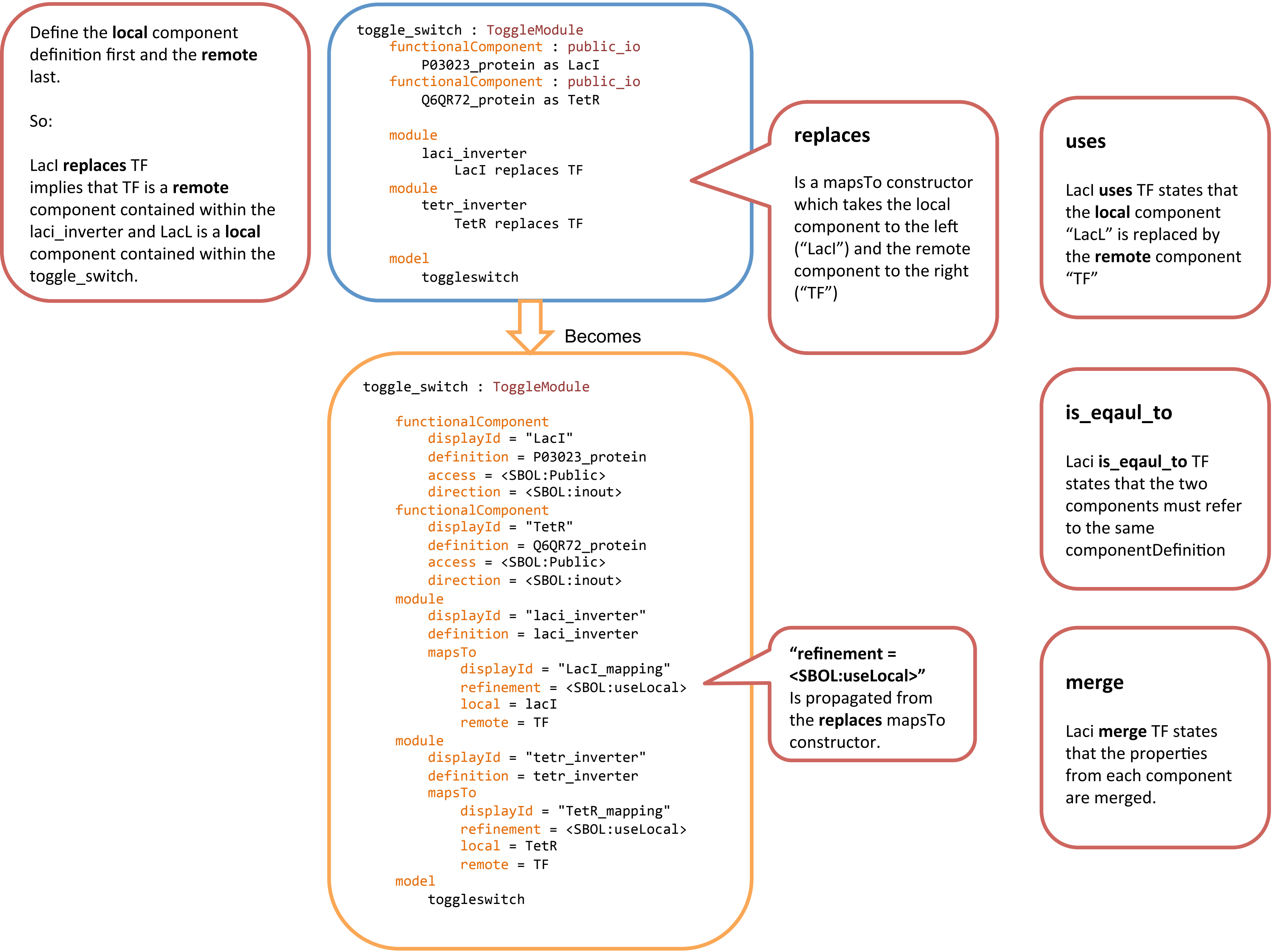

Define the **local** component definition first and the **remote** last.

So:

LacI **replaces** TF
implies that TF is a **remote** component contained within the laci_inverter and LacL is a **local** component contained within the toggle_switch.

```
toggle_switch : ToggleModule
    functionalComponent : public_io
        P03023_protein as LacI
    functionalComponent : public_io
        Q6QR72_protein as TetR

    module
        laci_inverter
            LacI replaces TF
    module
        tetr_inverter
            TetR replaces TF

    model
        toggleswitch
```

replaces

Is a mapsTo constructor which takes the local component to the left (“LacI”) and the remote component to the right (“TF”)

uses

LacI **uses** TF states that the **local** component “LacL” is replaced by the **remote** component “TF”

Becomes

```
toggle_switch : ToggleModule

    functionalComponent
        displayId = "LacI"
        definition = P03023_protein
        access = <SBOL:Public>
        direction = <SBOL:inout>
    functionalComponent
        displayId = "TetR"
        definition = Q6QR72_protein
        access = <SBOL:Public>
        direction = <SBOL:inout>
    module
        displayId = "laci_inverter"
        definition = laci_inverter
        mapsTo
            displayId = "LacI_mapping"
            refinement = <SBOL:useLocal>
            local = lacI
            remote = TF
    module
        displayId = "tetr_inverter"
        definition = tetr_inverter
        mapsTo
            displayId = "TetR_mapping"
            refinement = <SBOL:useLocal>
            local = TetR
            remote = TF
    model
        toggleswitch
```

is_eqaul_to

Laci **is_eqaul_to** TF states that the two components must refer to the same componentDefinition

“refinement = <SBOL:useLocal>”
Is propagated from the **replaces** mapsTo constructor.

merge

Laci **merge** TF states that the properties from each component are merged.