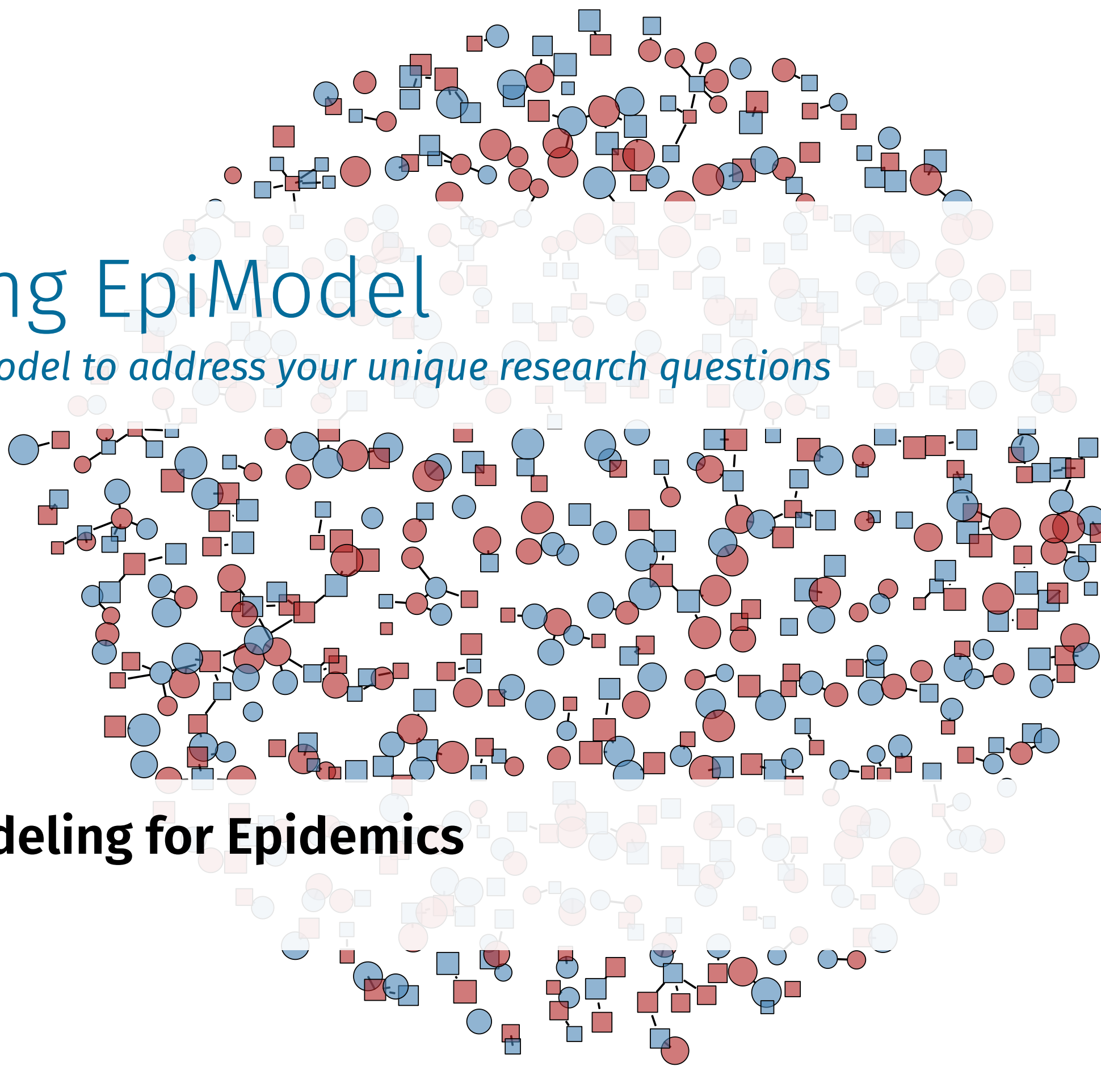




Extending EpiModel

How to use EpiModel to address your unique research questions

Network Modeling for Epidemics

Day 4

When you run netsim

```
netsim(est, param, init, control)
```

```
  t1 initialization module
```

```
  for (t in 2 to tn) {
```

```
    module 1
```

```
    module 2
```

```
    network re-simulation module
```

```
    transmission module
```

```
    module 5
```

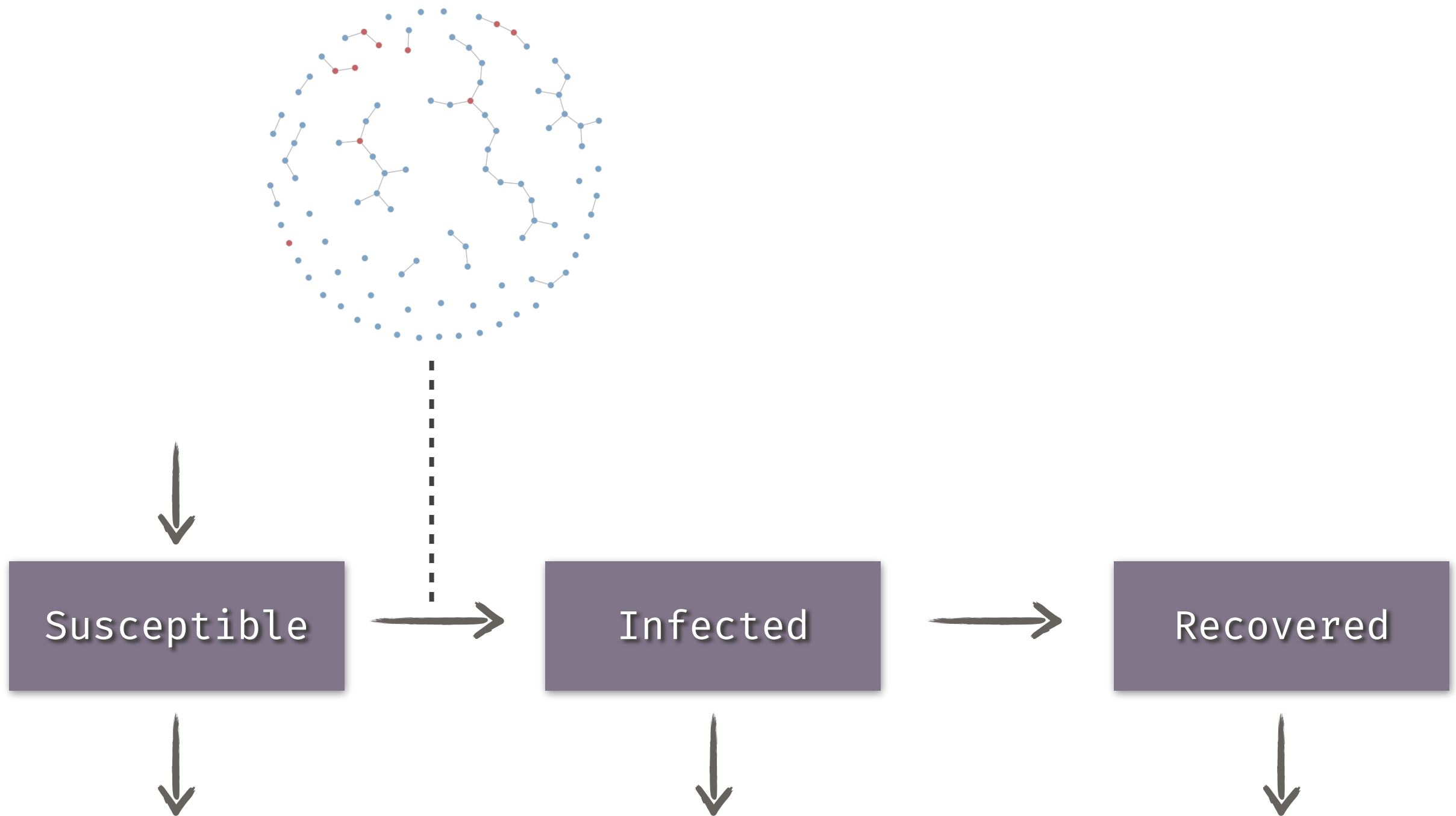
```
  }
```

```
  tn Clean up and save output
```

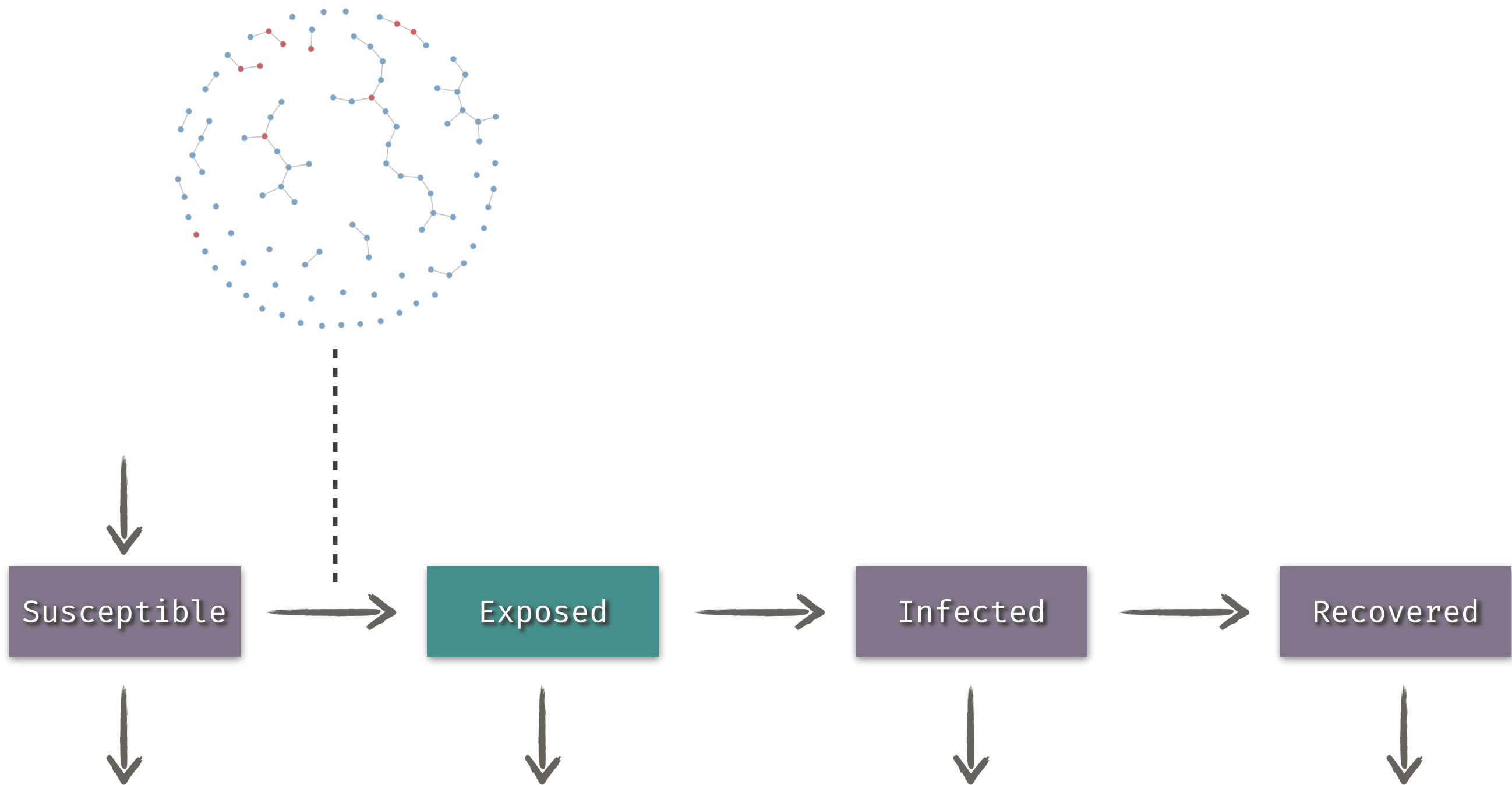
What are Modules?

- Perform **one** core process within the system
- Module = concept; module function = realization
- Inputs are data structure and parameters
- Outputs are **revised** data structure
- Standardized programming interface (API) for building modules
- “Plug and play” with EpiModel

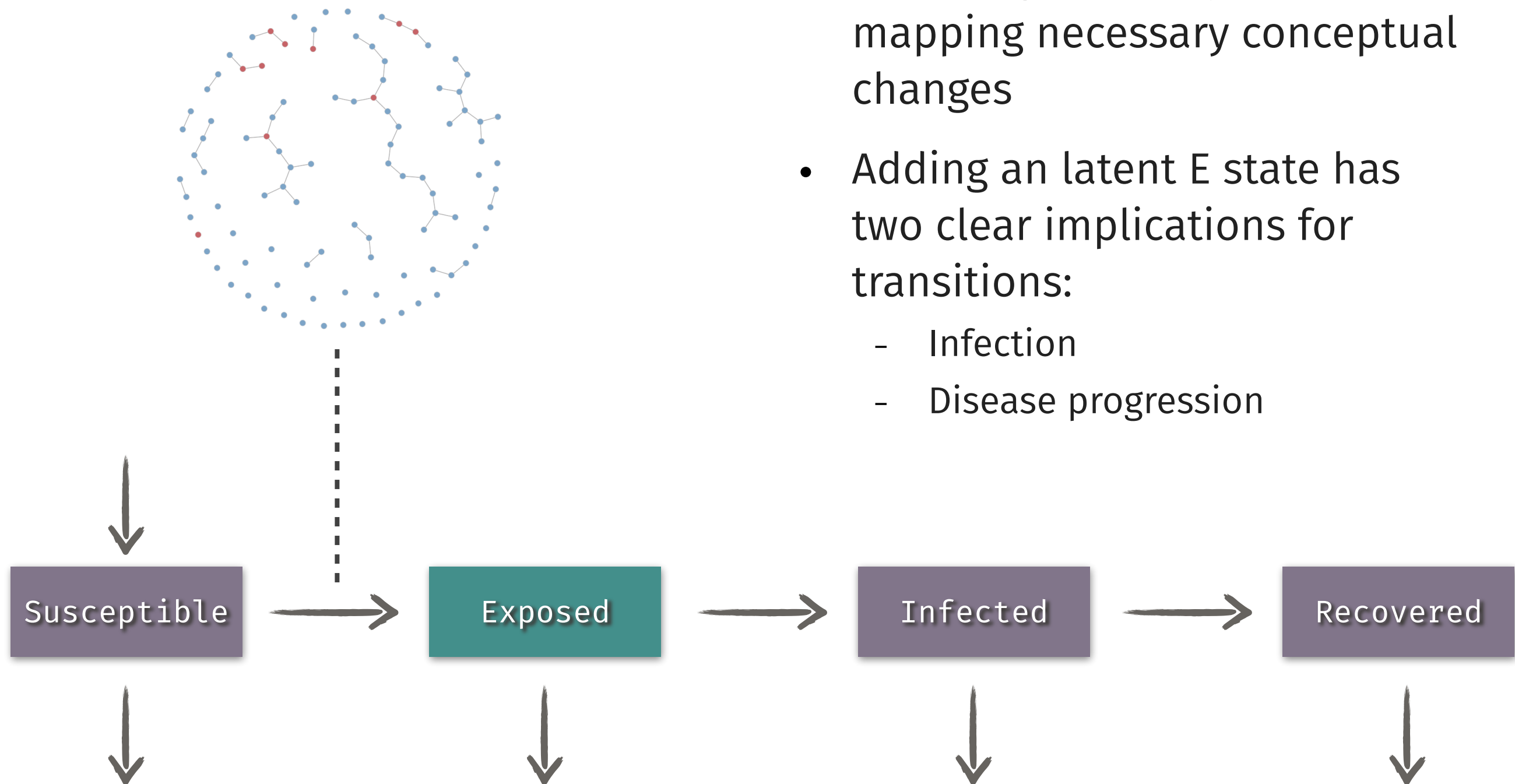
Concept $\Rightarrow$ Code



Concept $\Rightarrow$ Code



Concept $\Rightarrow$ Code

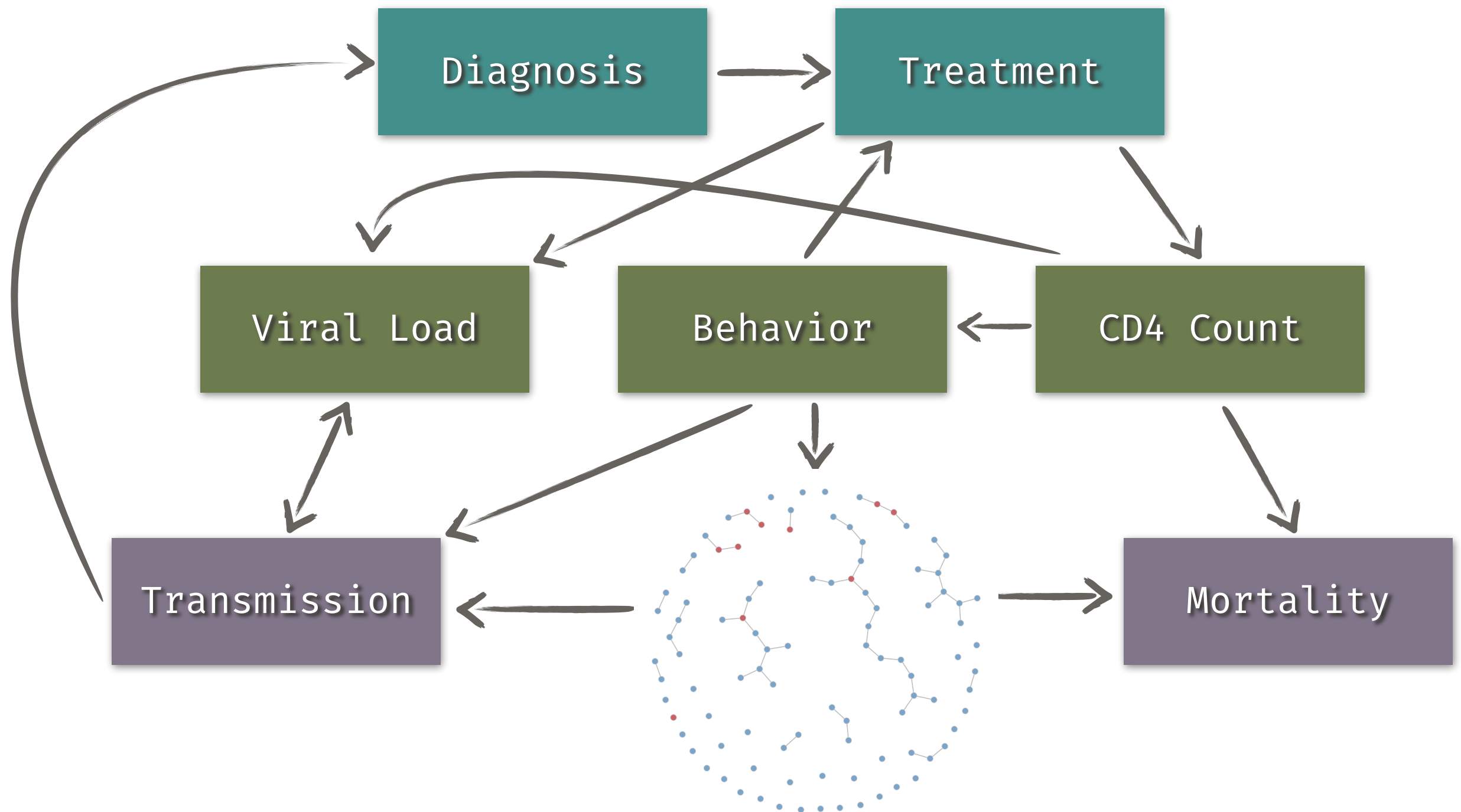


Built-In Modules

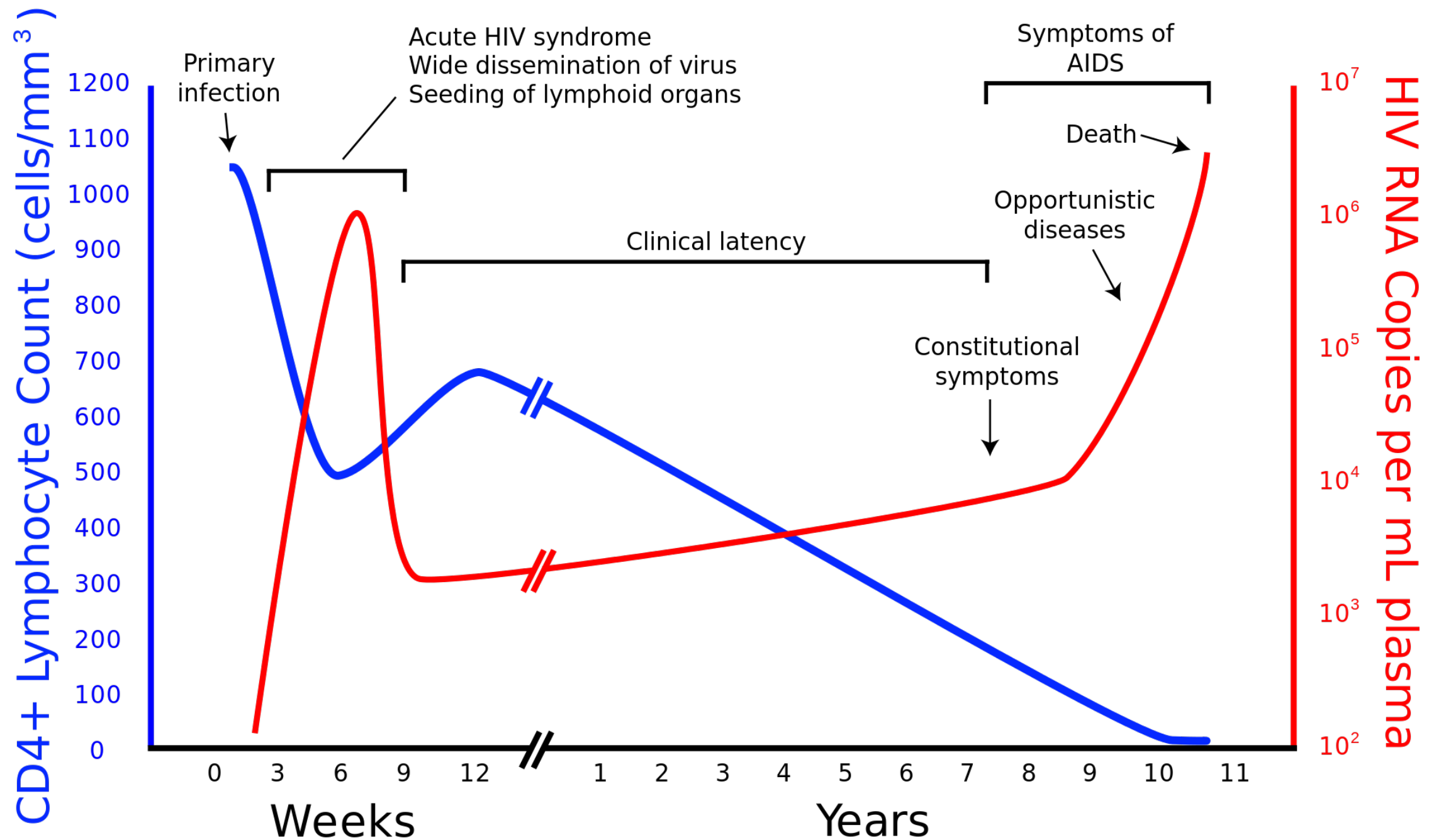
```
help("control.net")
```

<code>initialize.FUN</code>	Module to initialize the model at time 1, with the default function of <code>initialize.net</code> .
<code>departures.FUN</code>	Module to simulate death or exit, with the default function of <code>deaths.net</code> .
<code>arrivals.FUN</code>	Module to simulate births or entries, with the default function of <code>births.net</code> .
<code>recovery.FUN</code>	Module to simulate disease recovery, with the default function of <code>recovery.net</code> .
<code>resim_nets.FUN</code>	Module to resimulate the network at each time step, with the default function of <code>resim_nets</code> .
<code>infection.FUN</code>	Module to simulate disease infection, with the default function of <code>infection.net</code> .
<code>prevalence.FUN</code>	Module to calculate disease prevalence at each time step, with the default function of <code>get_prev.net</code> .
<code>verbose.FUN</code>	Module to print simulation progress to screen, with the default function of <code>verbose.net</code> .

Modular Approach Allows for Incremental Design of Complex System, with TERGM Machinery Central



Module Complexity Should Reflect Research Question



EpiModelHIV

- Extension modules for EpiModel specifically for modeling HIV infection
 - Modules geared towards heterosexual transmission in Sub-Saharan Africa and MSM in the United States
- Modules include:
 - Natural disease progression impact on CD4 and HIV viral load trajectories
 - Inter-host transmission risk dependent on disease stage and VL
 - Anti-retroviral therapy treatment

<https://github.com/statnet/EpiModelHIV>

EpiModelHIV Modules

aging.FUN	Module function for aging.
deaths.FUN	Module function for general and disease-related deaths.
births.FUN	Module function for births or entries into the population.
test.FUN	Module function for diagnostic disease testing.
tx.FUN	Module function for ART initiation and adherence.
prep.FUN	Module function for PrEP initiation and utilization.
progress.FUN	Module function for HIV disease progression.
v1.FUN	Module function for HIV viral load evolution.
aiclass.FUN	Module function for one-off AI risk class transitions.
roleclass.FUN	Module function for transitions in sexual roles.
edgescorr.FUN	Module function for the edges coefficient adjustment to preserve mean degree
resimnets.FUN	Module function for network resimulation at each time step.
disclose.FUN	Module function for HIV status disclosure.
acts.FUN	Module function to simulate the number of sexual acts within partnerships.
condoms.FUN	Module function to simulate condom use within acts.
riskhist.FUN	Module function to calculate risk history for uninfected persons in the population.
position.FUN	Module function to simulate sexual position within acts.
trans.FUN	Module function to stochastically simulate disease transmission
getprev.FUN	Module function to calculate prevalence summary statistics.
verbose.FUN	Module function to print model progress to the console or external text files.

The EpiModel Extension API

1. Modules have associated function with standard inputs and outputs
 - Inputs are `dat` and `at`, outputs are `dat`
2. Modules use the accessor `get_/set_` functions to read and write sublist data to the `dat` object; See `help("net-accessor")`
 - epi summary stats updated with single value defined for current time step
 - nodal attributes updated, with entry for all nodes on network
3. Modules defined and called into `EpiModel::netsim` through `control.net` settings
 - Each defined module parameter ends in `.FUN`
 - `type` control setting should be set to `NULL` for any extension models
 - Standard modules — handling `dat` initialization, network resimulation, and network data updates — are not intended to be edited by end users (but may be as necessary)

The EpiModel Extension API

4. Nodal attributes may be initialized on the network as prior to TERGM estimation with `set_vertex_attribute`
 - Applies even if attributes are not called during TERGM estimation
5. Models with departures (e.g., mortality) must update nodal attributes
 - `active` must be set to `0` for all departing nodes
 - `exitTime` must be set to `at` for all departing nodes
6. Models with arrivals (e.g., births) must append nodal attributes for incoming nodes
 - `append_attr` used in arrival module to set new nodal attributes