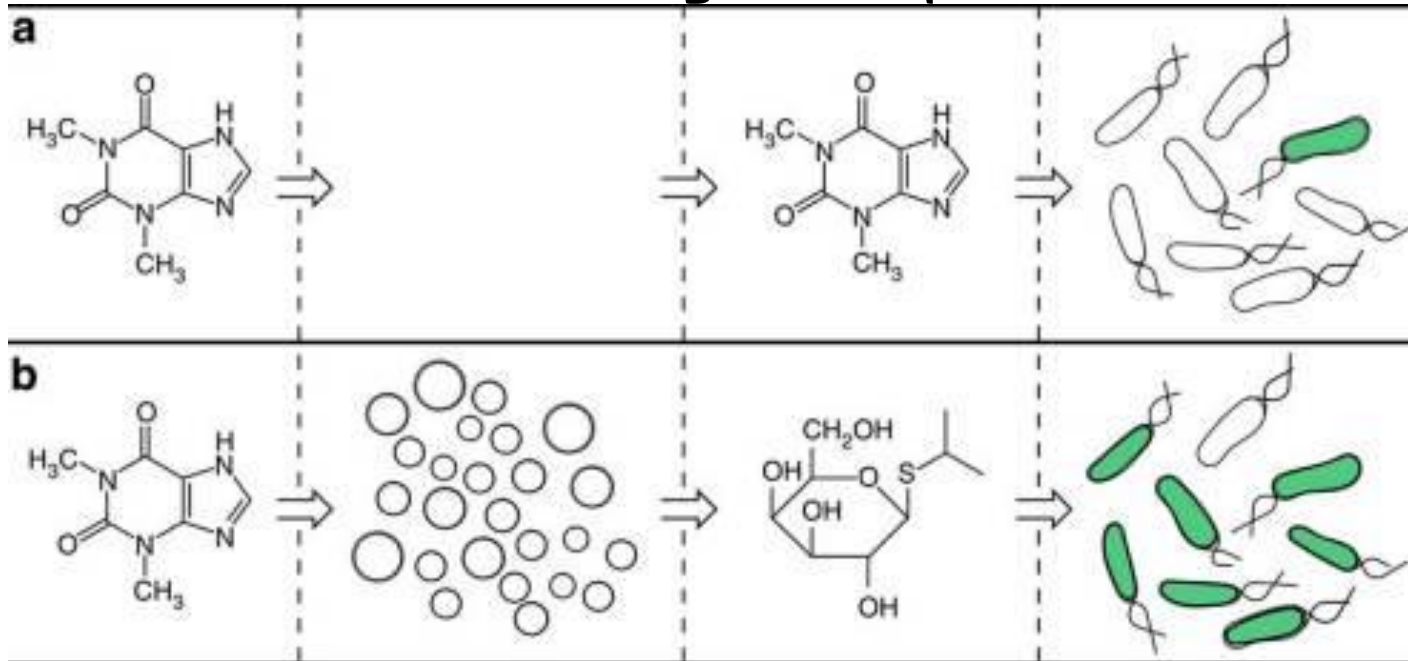


# Sub-SBML: A Subsystem Interaction Modeling Python Toolbox for SBML Models

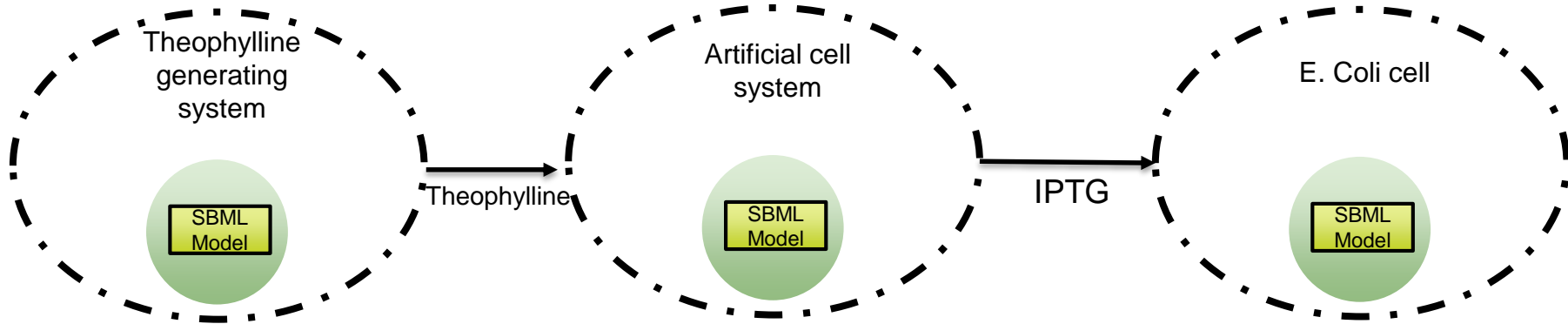
Ayush Pandey  
Murray Lab, Caltech

# A demonstrative modeling example



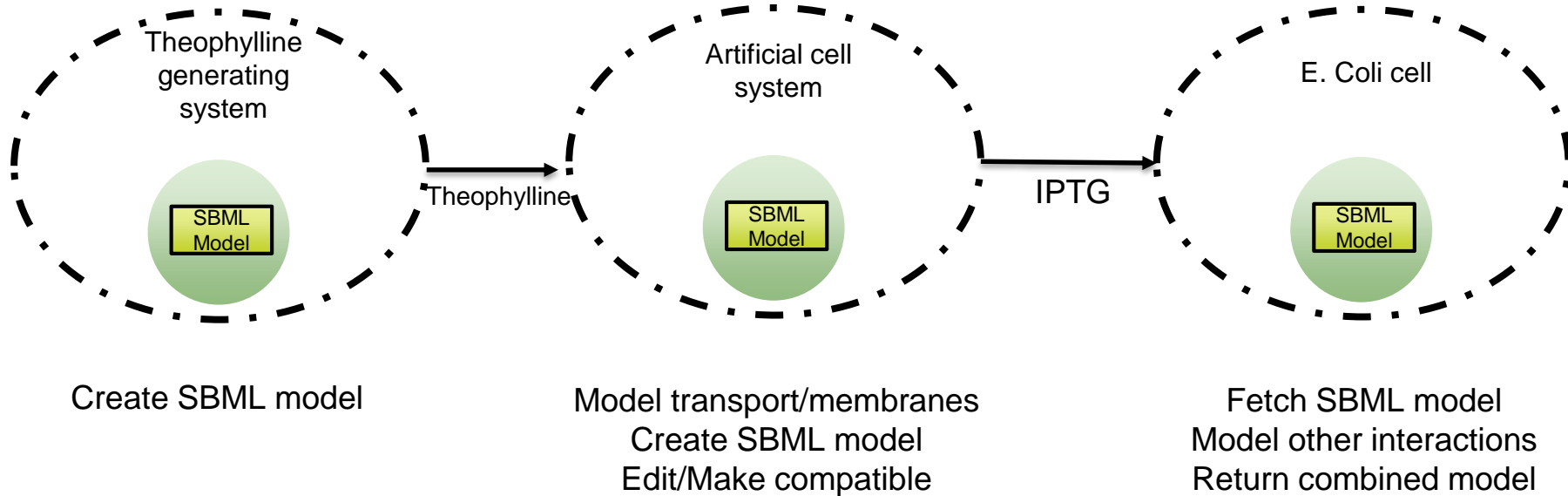
Reference : "Integrating artificial with natural cells to translate chemical messages that direct *E. coli* behavior", Roberta Lentini, Silvia Perez Santero et al. In: *Nature Communications*, 2014

# How to model this system?

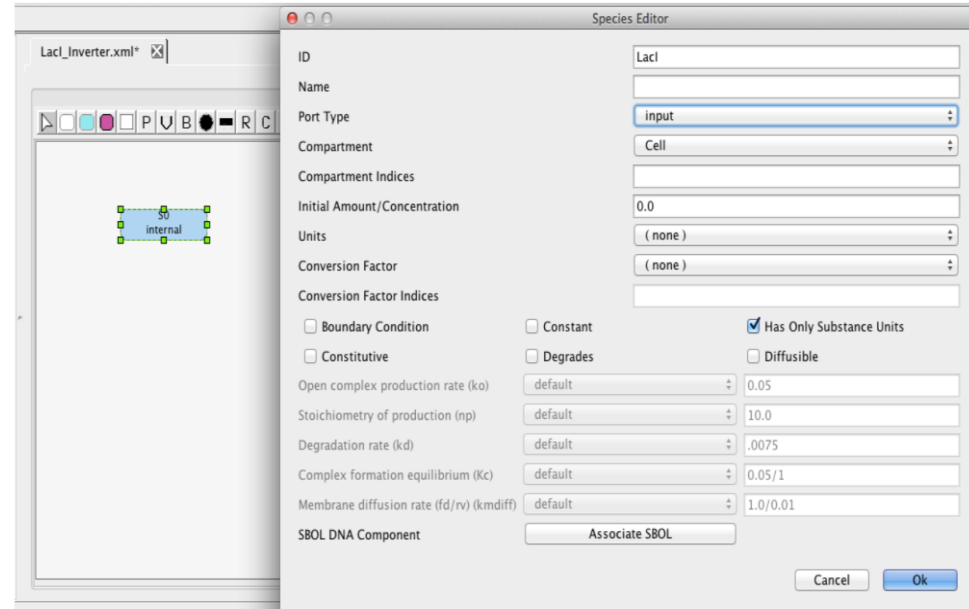
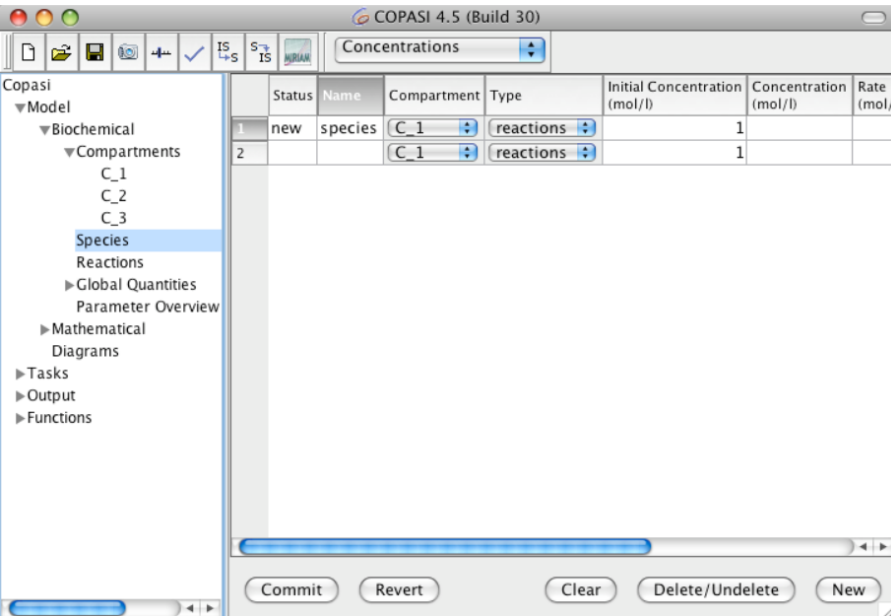


Reference : "Integrating artificial with natural cells to translate chemical messages that direct *E. coli* behavior", Roberta Lentini, Silvia Perez Santero et al. In: *Nature Communications*, 2014

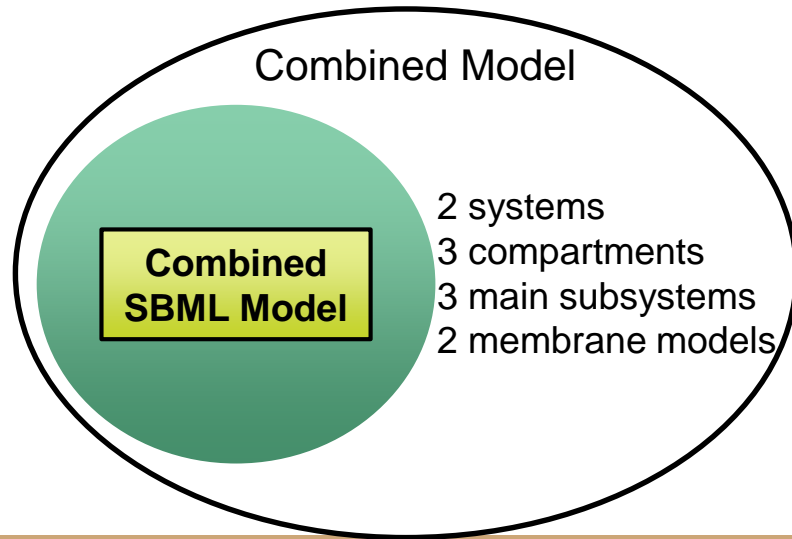
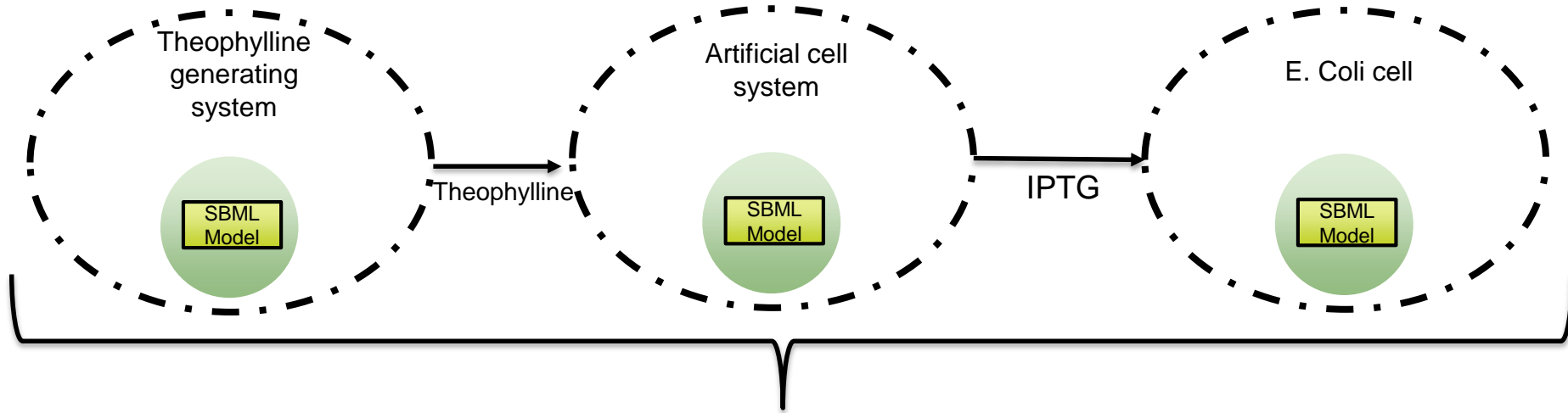
# How to model this system?

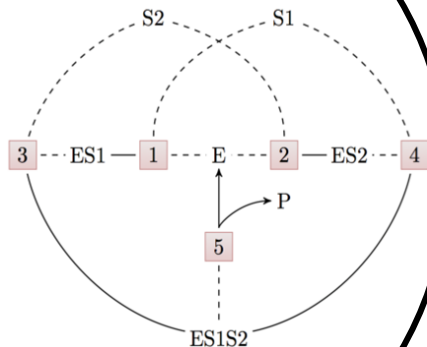
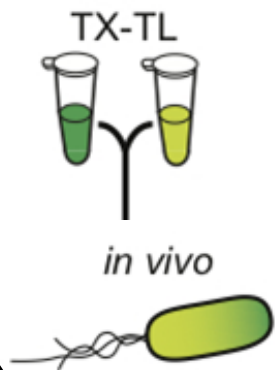


# Using existing tools – COPASI, iBioSim

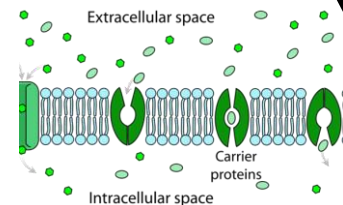
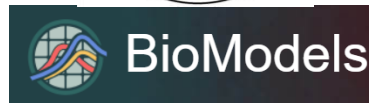
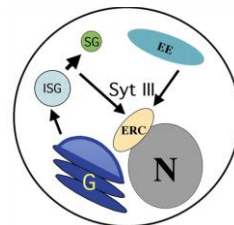


Using Sub-SBML...





**Create Models**



**Compile Models**

Model Reduction

**Analysis with Models**

Sensitivity analysis

System identification

Guiding experiments

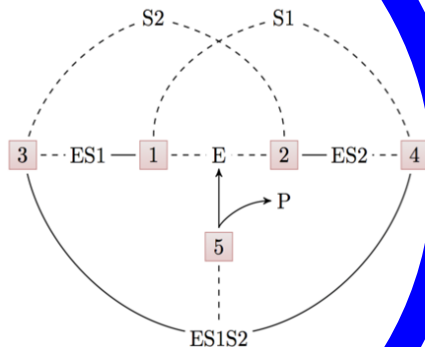




TX-TL



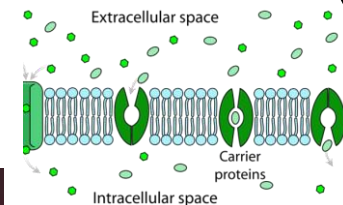
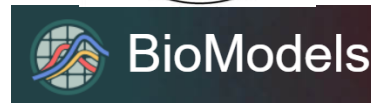
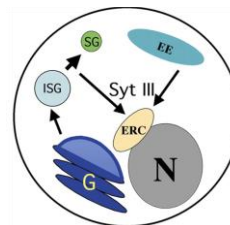
*in vivo*



**Create Models**



libSBML



**Compile Models**

Model Reduction

**Analysis with Models**

Sensitivity analysis

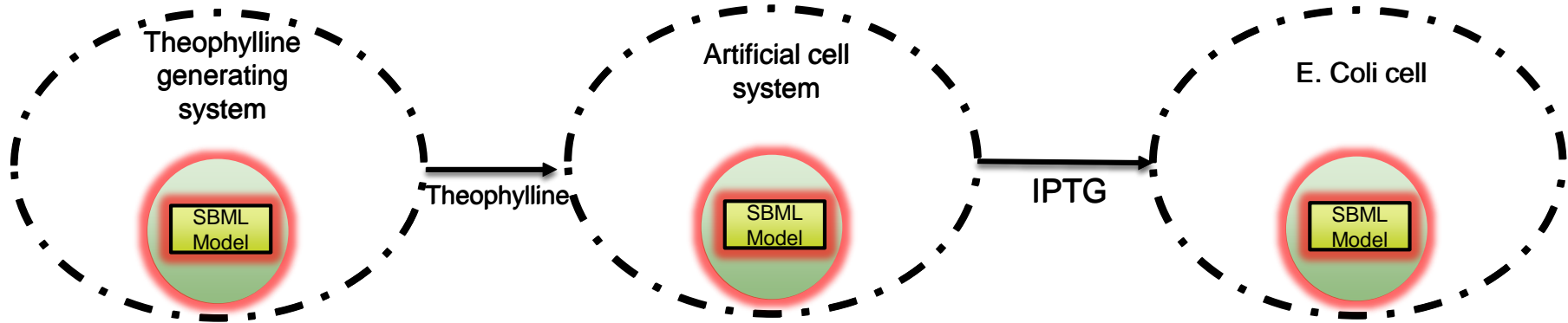
System identification

Guiding experiments

# Create Models

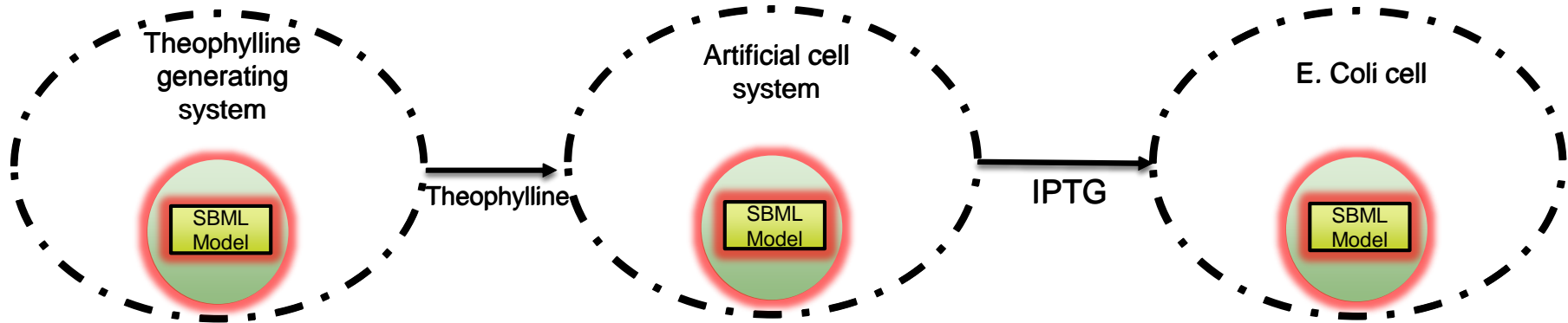
- Creating SBML models
  - Use Sub-SBML
  - ... or any other existing tool

# Create SBML Subsystems



1. Create a new Subsystem
2. Load SBML file into subsystem / Create SBML components from scratch
3. Use libsbml API directly or simple commands in subsbml

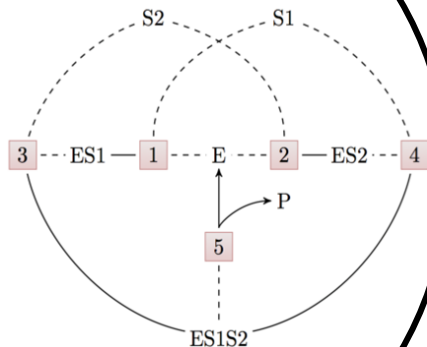
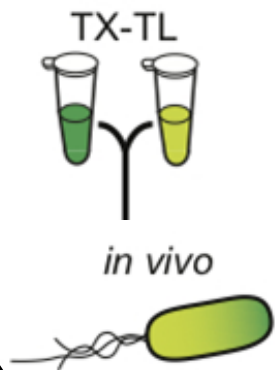
# Create SBML Subsystems



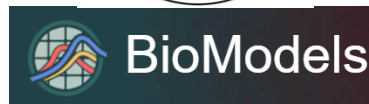
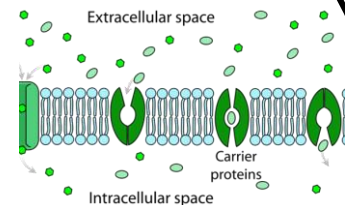
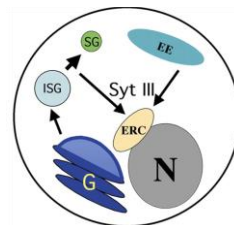
```
artificial_cell = createNewSubsystem(), model = artificial_cell.getModel()  
model.createNewSpecies(ListOfSpecies, ListofAmounts, ... ) ...
```

OR

```
artificial_cell = createSubsystem('load_model.xml')
```



**Create Models**



**Compile Models**

Model Reduction

**Analysis with Models**

Sensitivity analysis

System identification

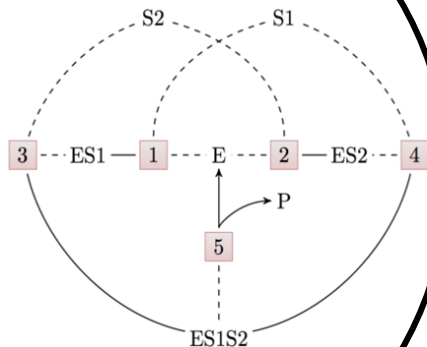
Guiding experiments



TX-TL

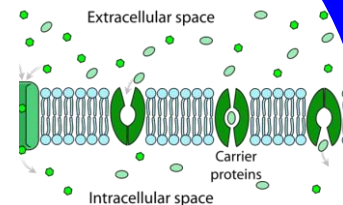
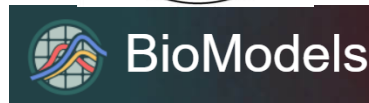
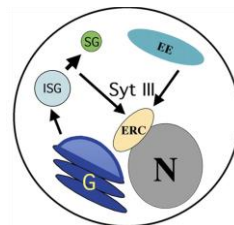


*in vivo*



**Create Models**

SBML libSBML



**Compile Models**

Model Reduction

**Analysis with Models**

Sensitivity analysis

System identification

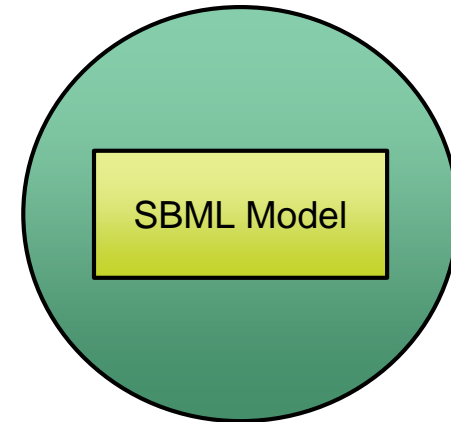
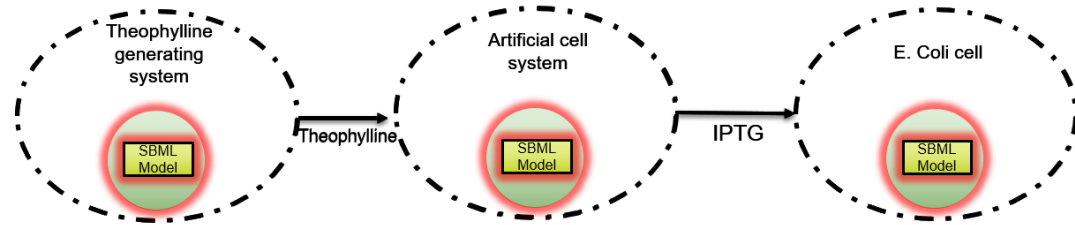
Guiding experiments

# Compile Models

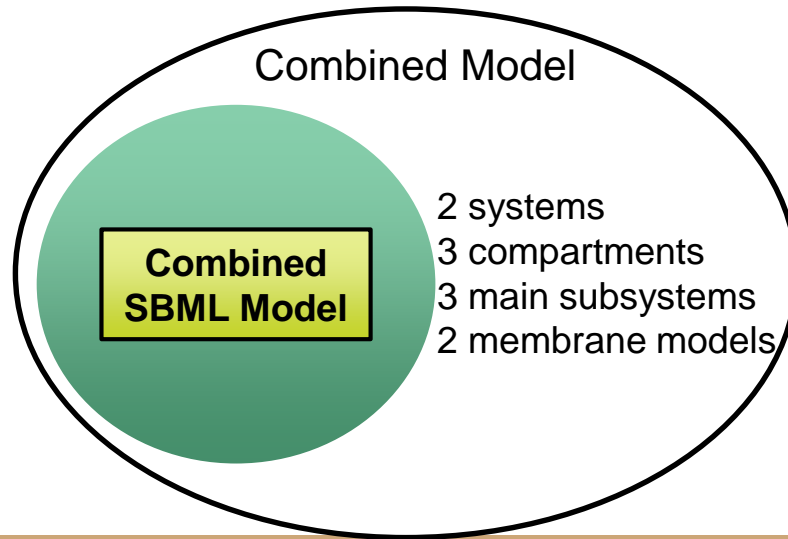
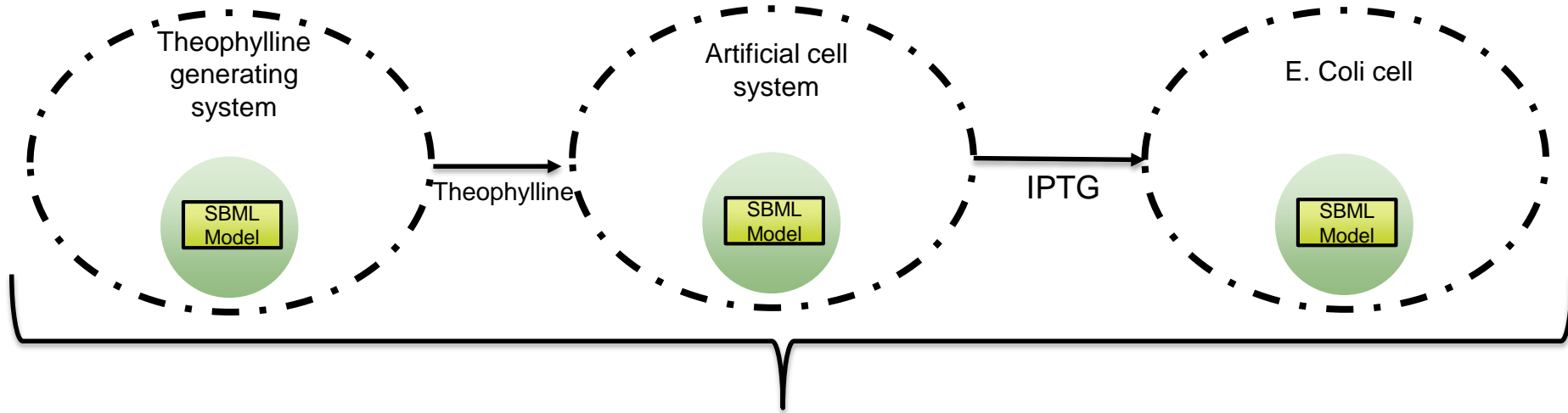
- Utility of Subsystems
- Combining multiple SBML models
- Transport and compartment modeling
  - Interactions between different compartments
  - Modeling of transport across membranes

# Modeling Subsystems

- Contains one SBML model
  - 1 compartment model
- Various utility functions available for Subsystems to edit models
  - Compiling multiple models
  - Interaction modeling
  - Simulating variable inputs
  - Analysis tools
- Other applications can be built on top of Subsystem architecture of modeling







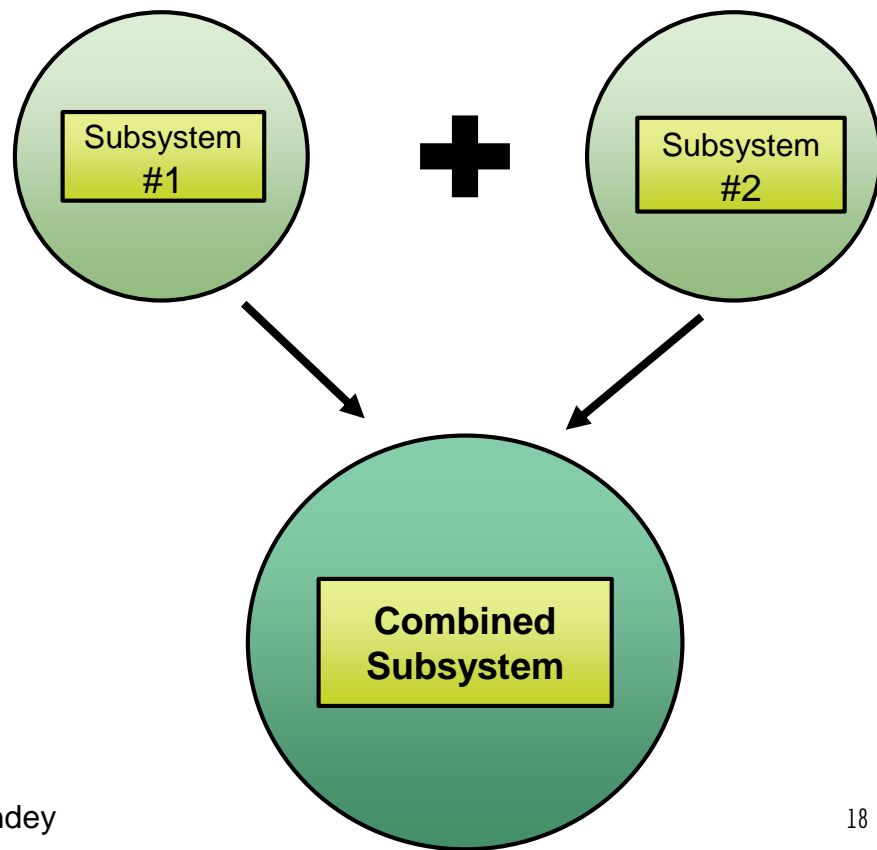
# Combining Subsystems

- What about common components?
- Shared resources?
- Duplicate species?
- Duplicate reactions?
- Other interactions?

Sub-SBML provides

- *shareSubsystems*
- *combineSubsystems*
- *connectSubsystems*

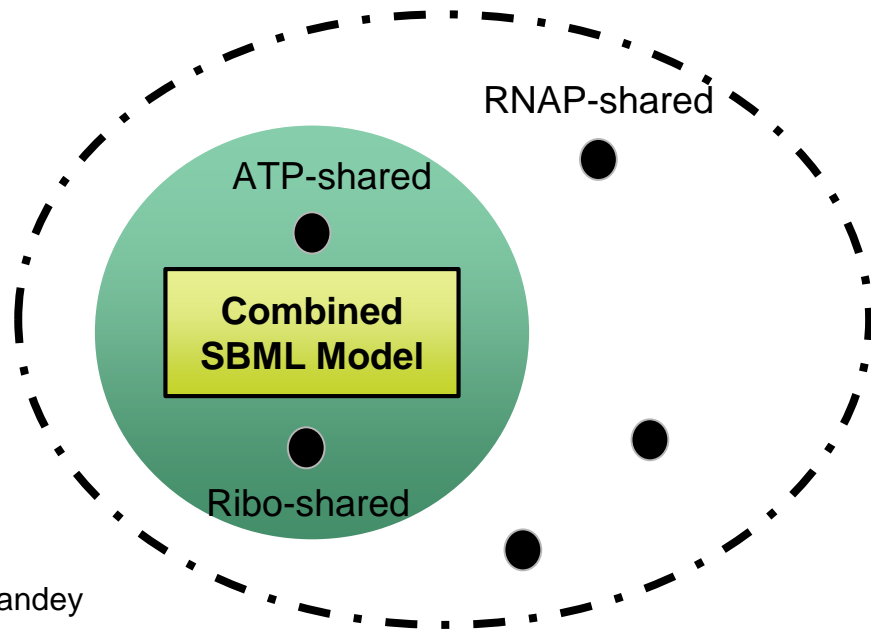
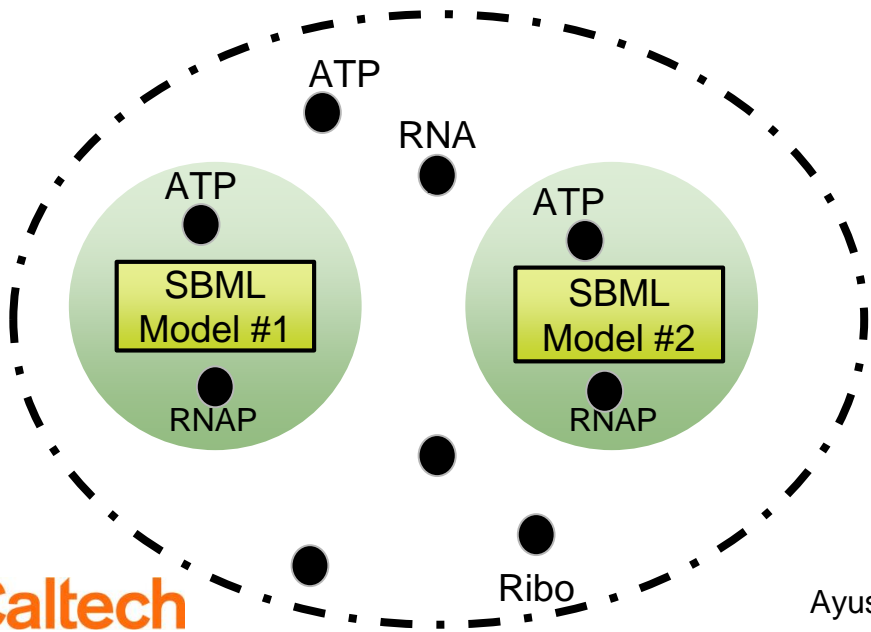
And builds other tools using these....



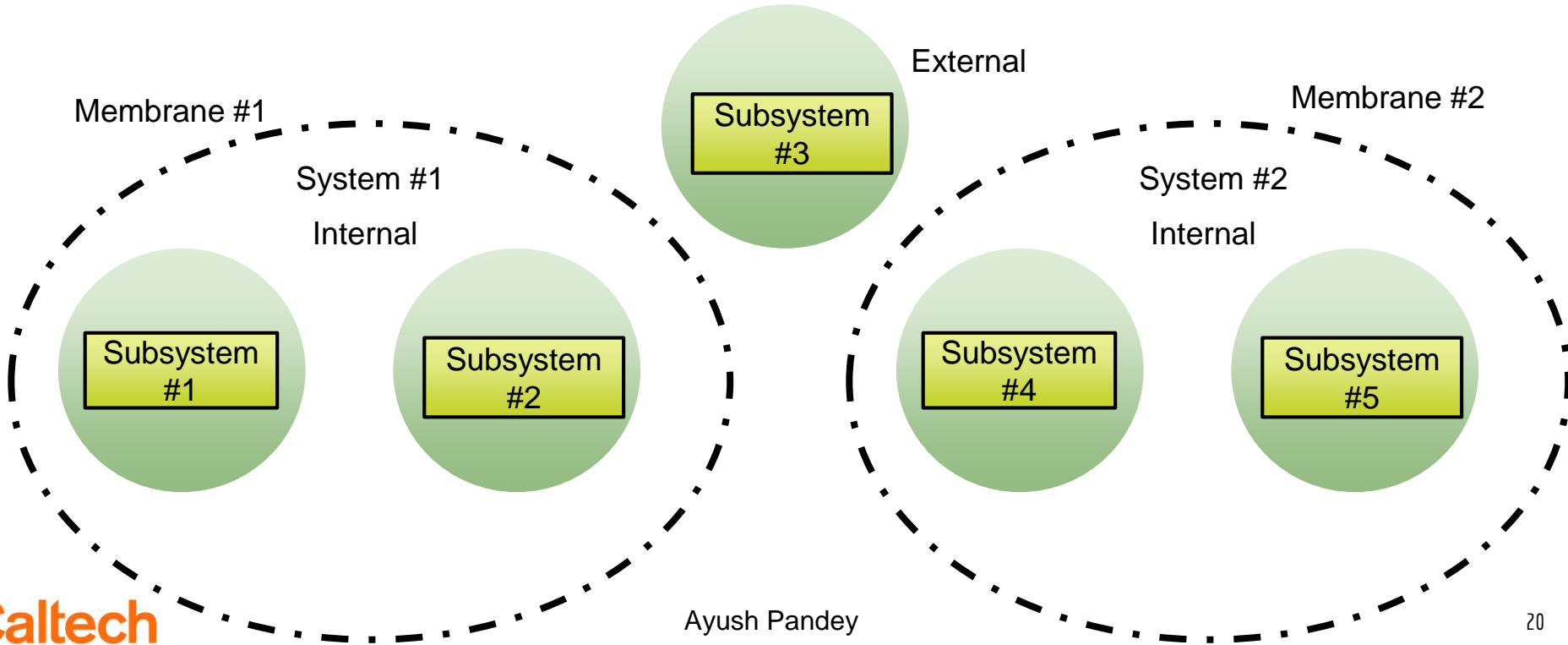
# Modeling : Shared Resources

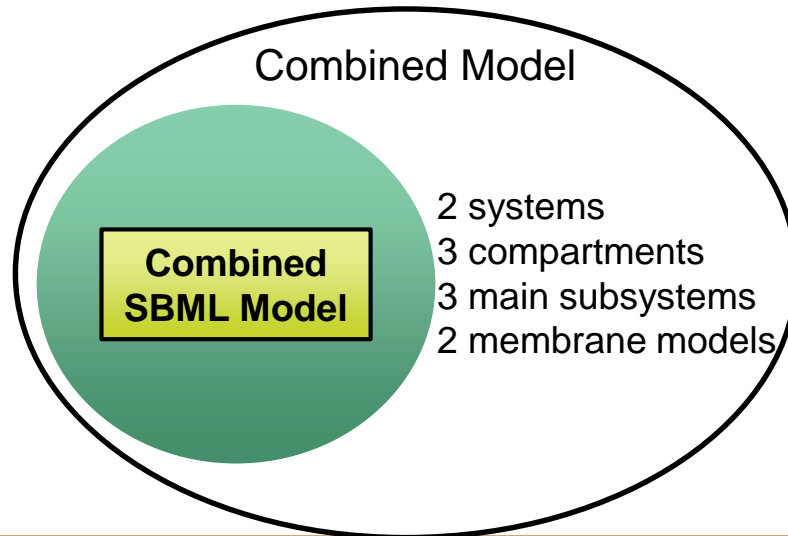
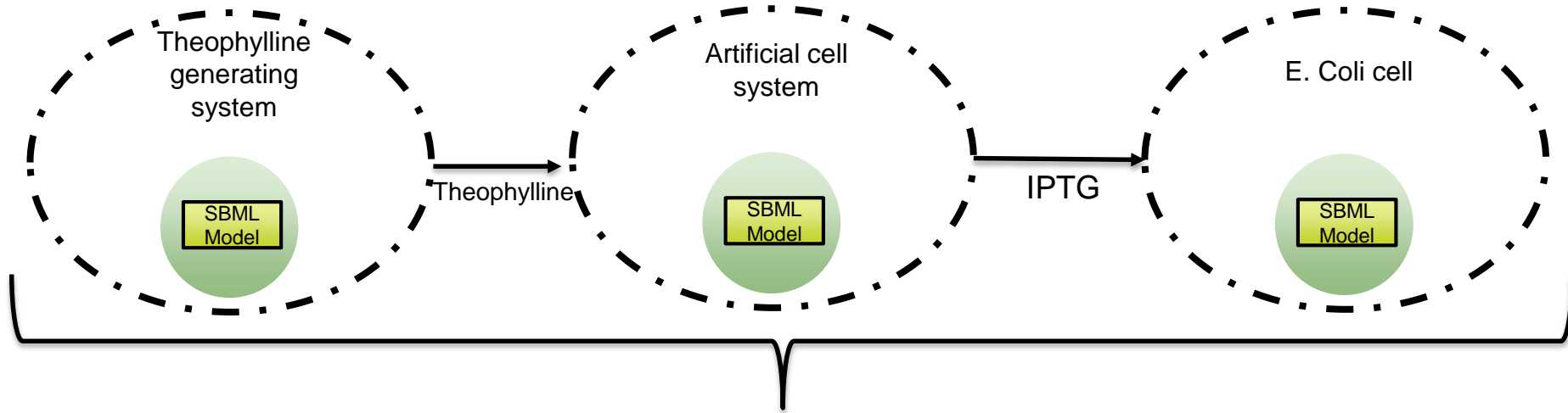
Usage -

```
cell = System()  
cell.createSubsystem('SS1.xml'), cell.createSubsystem('SS2.xml')  
shared_model = cell.setSharedResources(['ATP', 'RNAP', 'Ribo'])
```

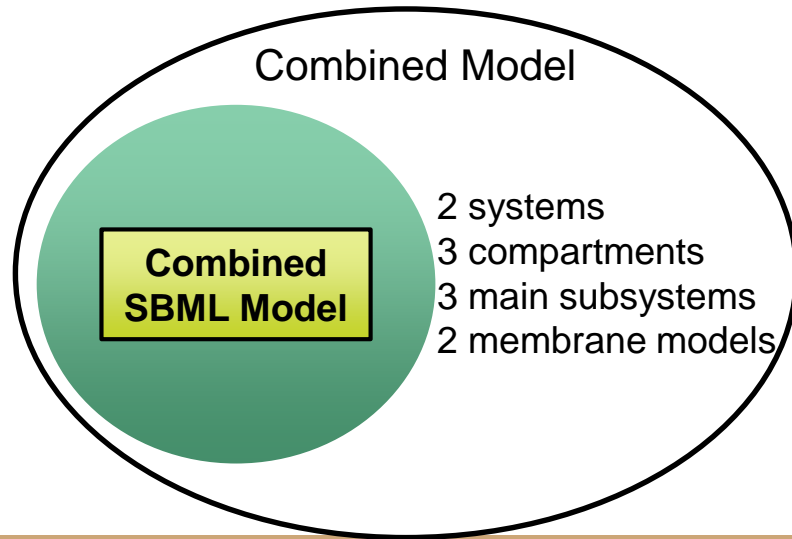
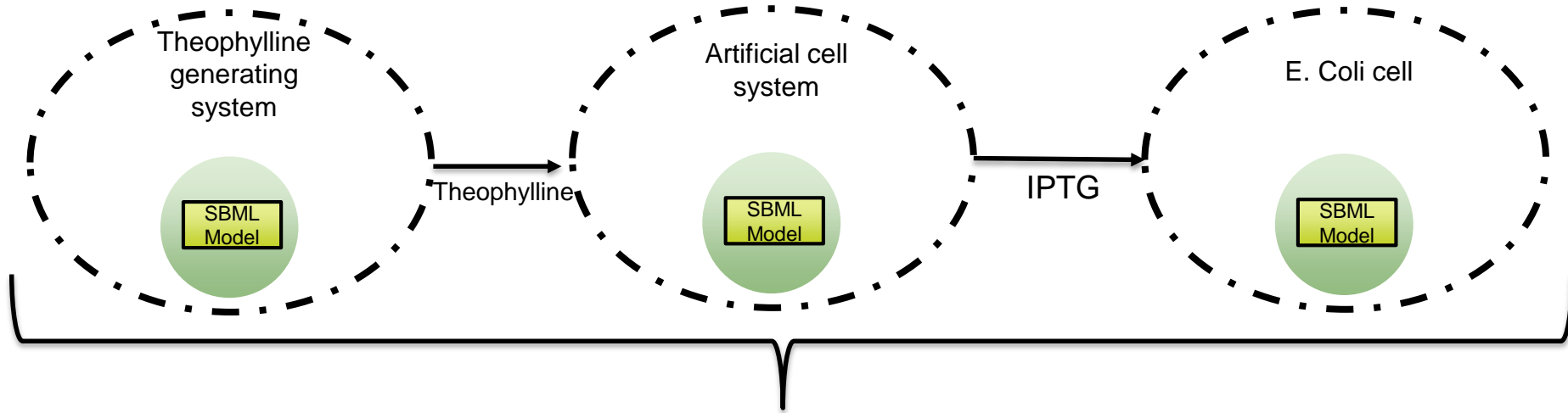


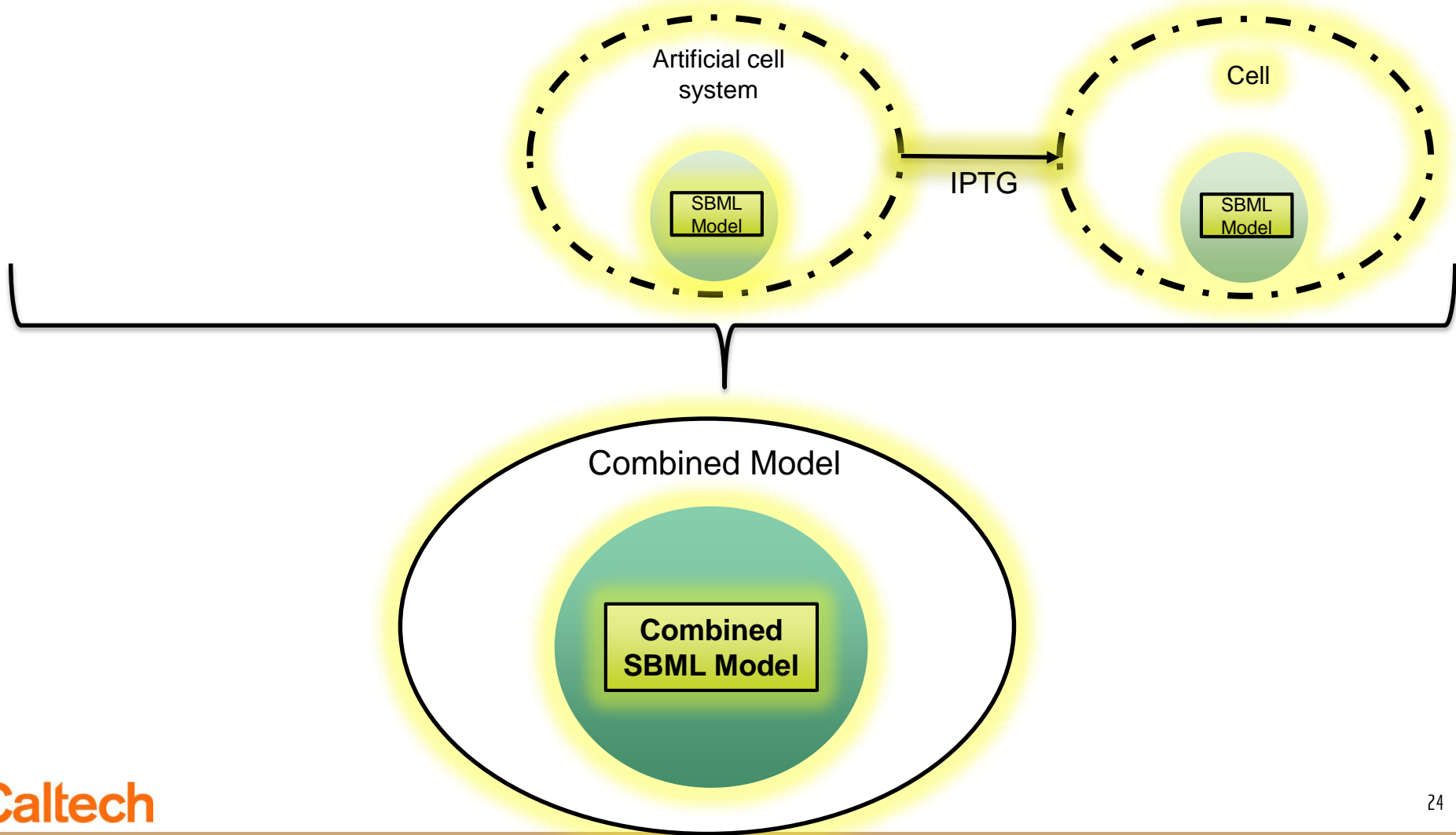
# “Subsystems” inside a “System”



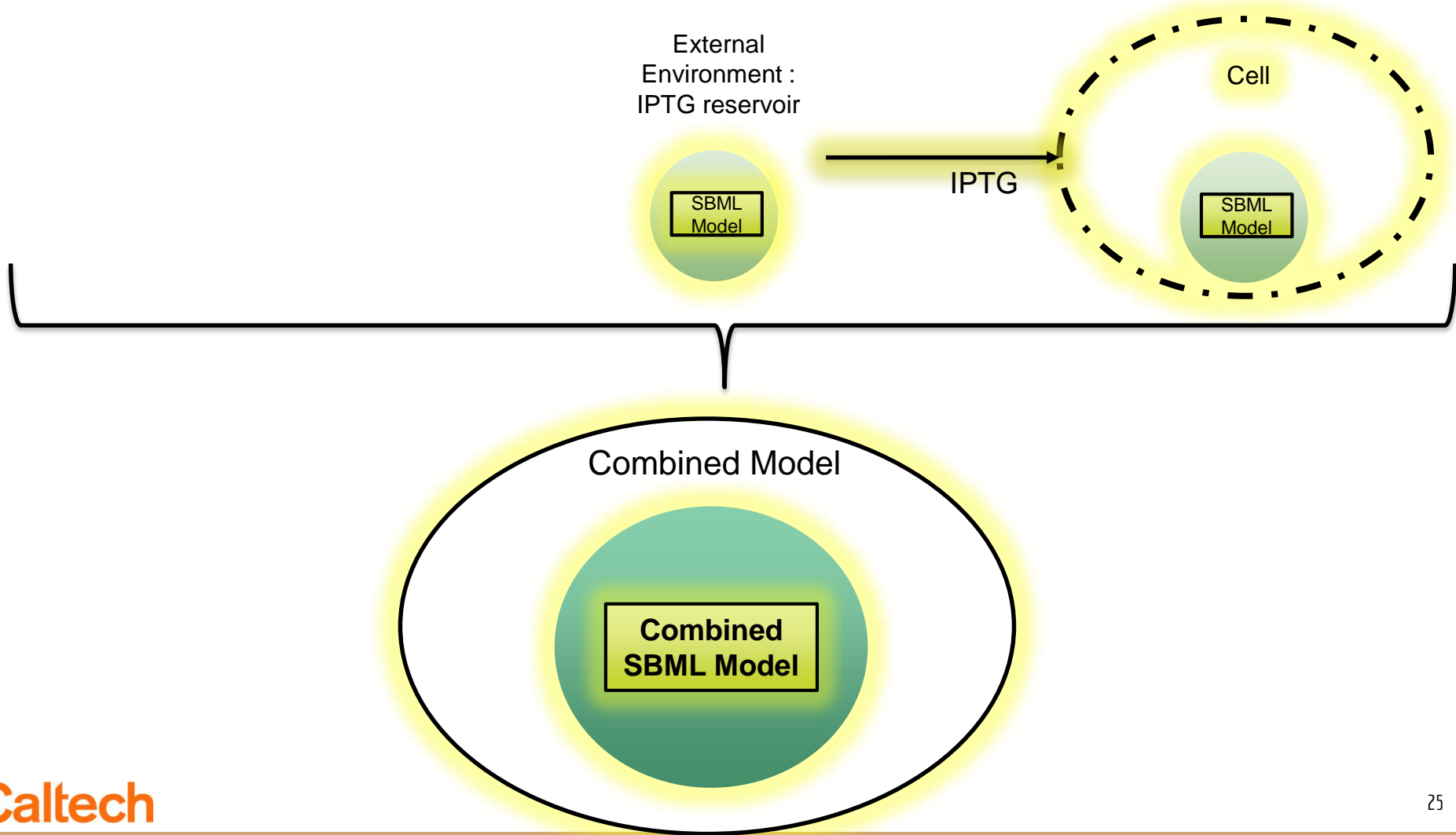


Considering only IPTG  
transport module...



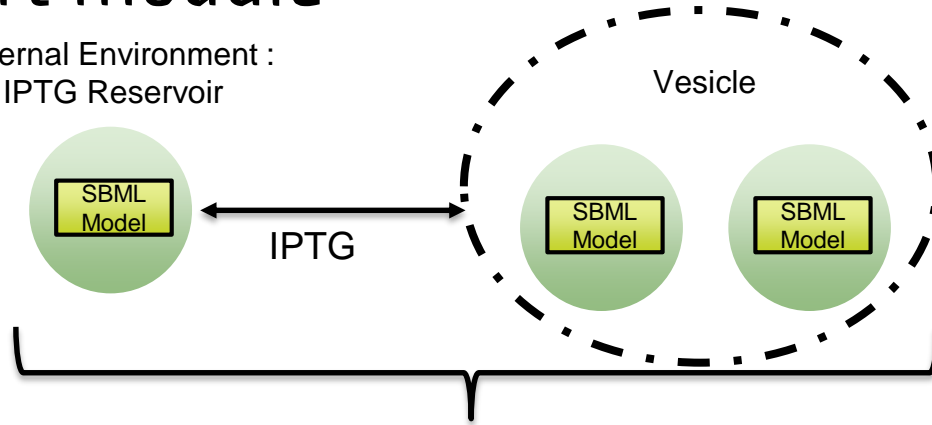




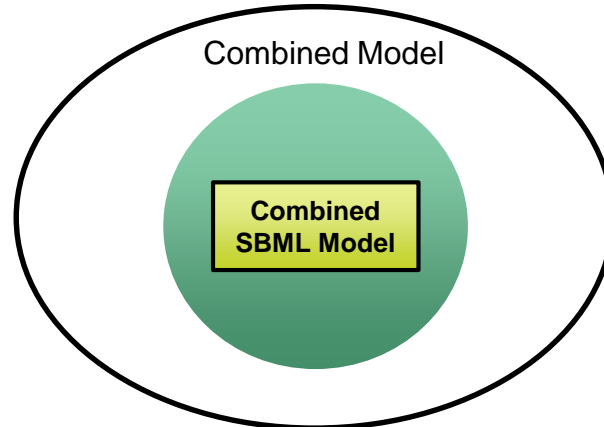


# IPTG transport module

External Environment :  
IPTG Reservoir



Combined Model



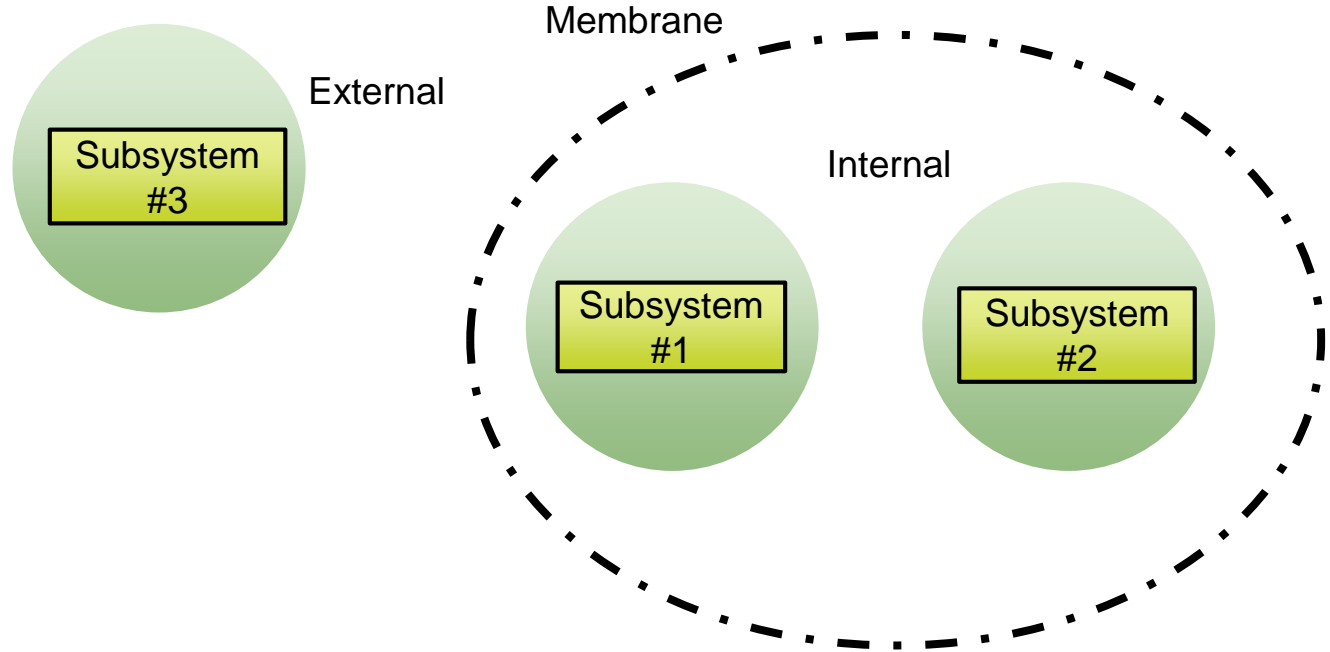
# IPTG Transport Modeling

Subsystem 1 -  
pOR1OR2-alpha-  
Hemolysin

Subsystem 2 -  
pLac-deGFP  
pOR1OR2-LacI

External -  
IPTG reservoir

Membrane -  
Allows IPTG  
transport

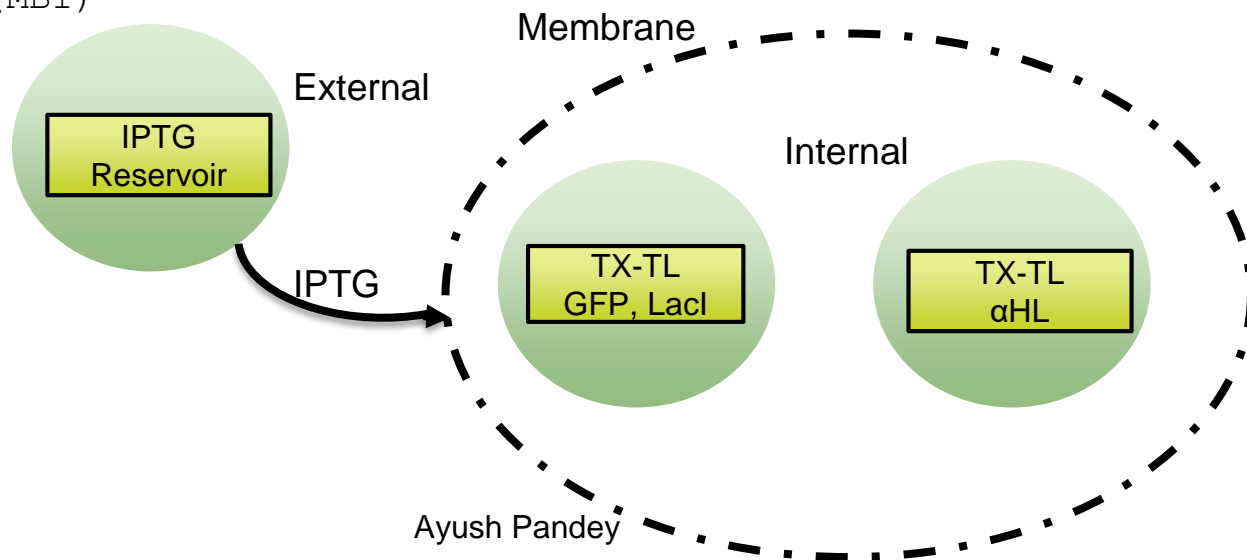


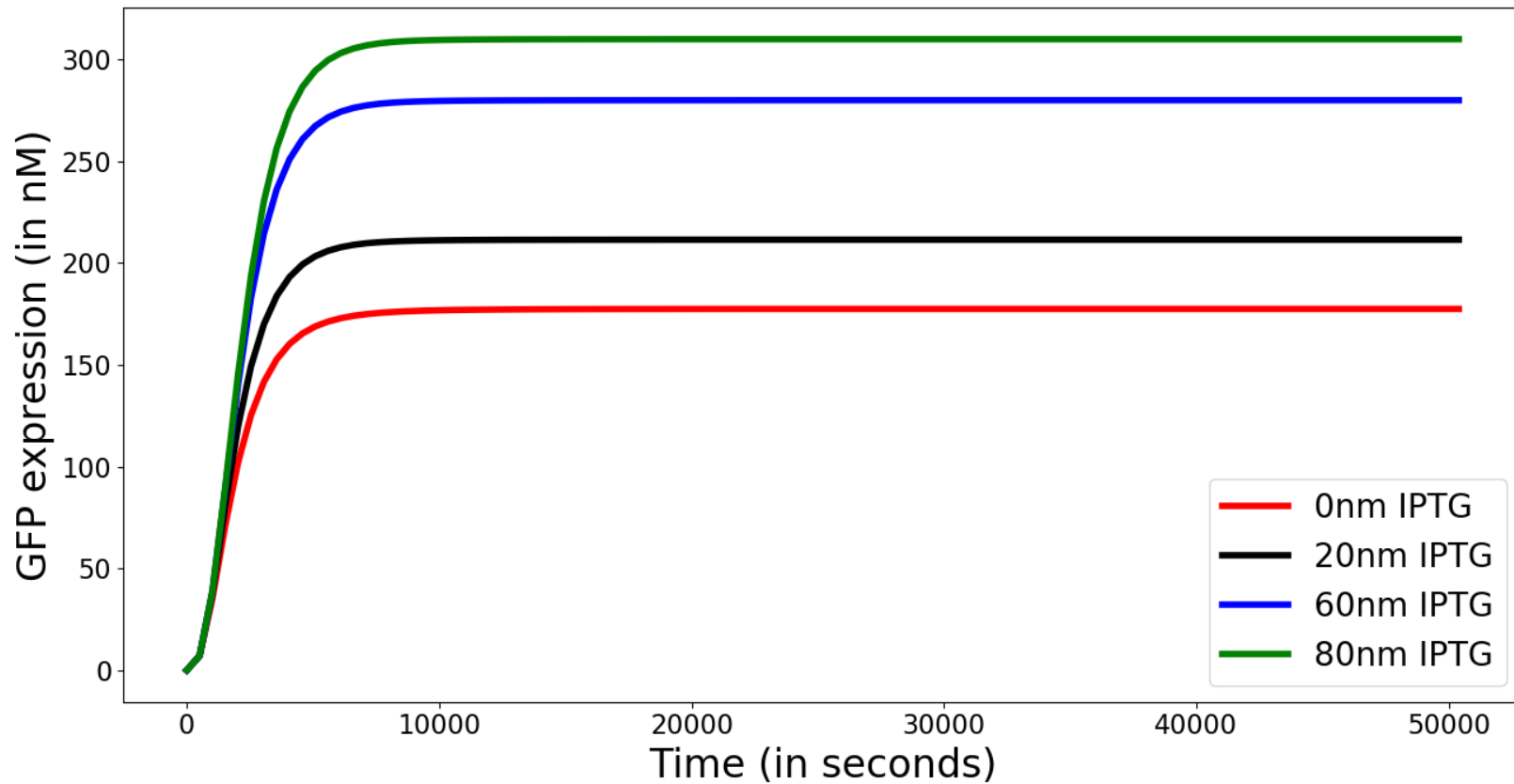
# Modeling using Sub-SBML

Modeling -

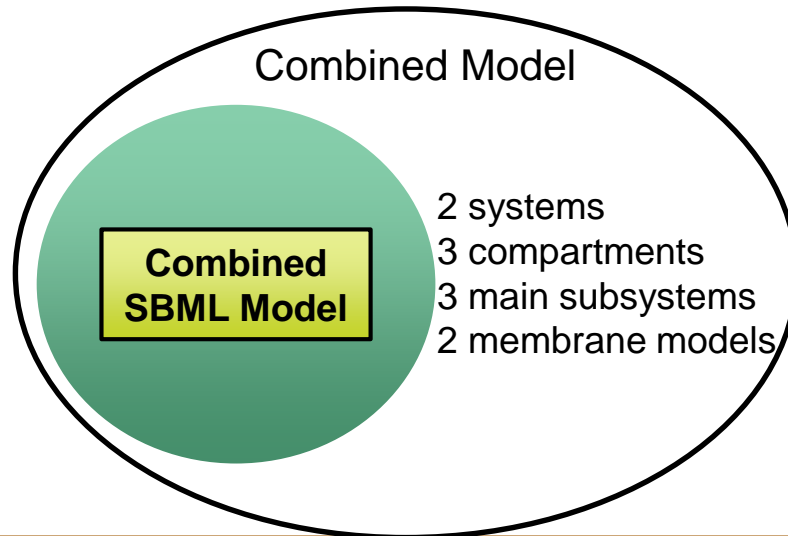
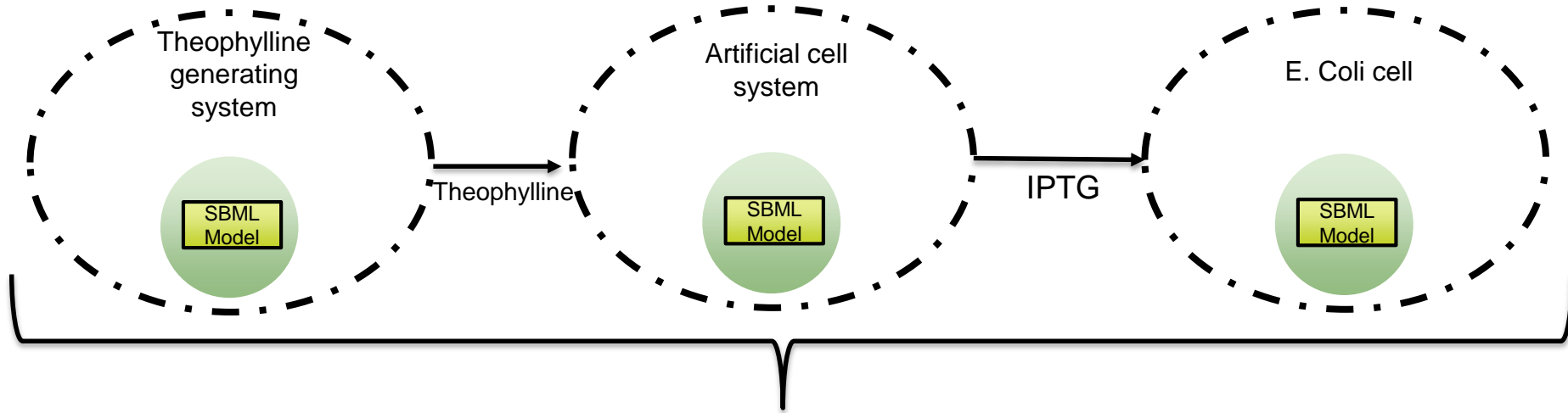
```
sys1 = System(),  
SS1 = sys1.createSubsystem('SS1.xml'), ...and so on  
sys1.setInternal([SS1, SS2])  
sys1.setExternal(SS3)  
sys1.setMembrane(MB1)
```

model = sys1.getModel()





Now including all other  
modules...



# Code in Sub-SBML

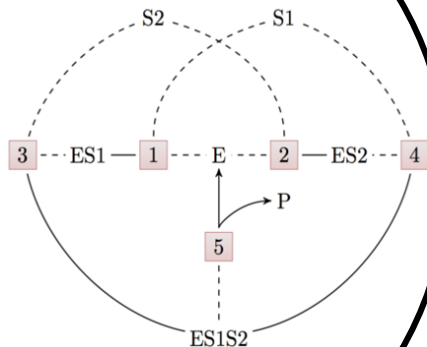
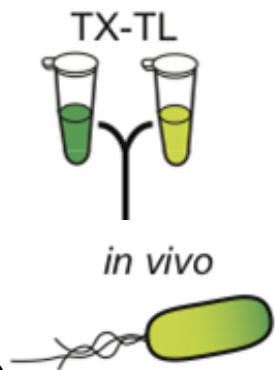
```
ss1 = createSubsystem('theophylline.xml')
ss2 = createSubsystem('artificial_cell.xml')
ss3 = createSubsystem('e_coli.xml')

mb1 = createSubsystem('membrane_artificial.xml')
mb2 = createSubsystem('membrane_IPTG.xml')

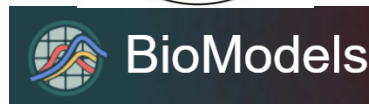
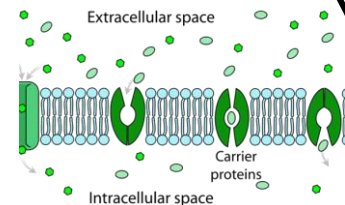
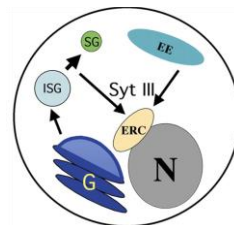
sys1 = System('cell1', [ss1,ss2], mb1)
sys2 = System('cell2', ss3, mb2)

combined_model = combineSystems(sys1.getModel(), sys2.getModel())
```





**Create Models**



**Compile Models**

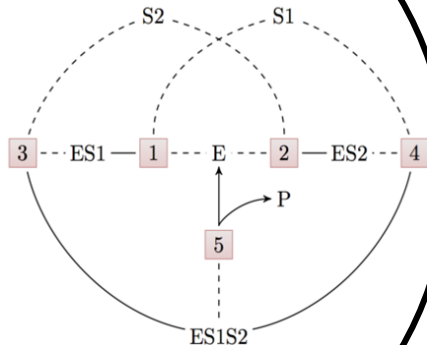
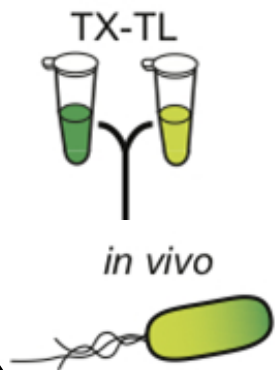
Model Reduction

**Analysis with Models**

Sensitivity analysis

System identification

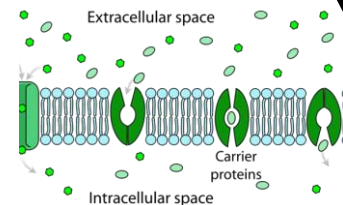
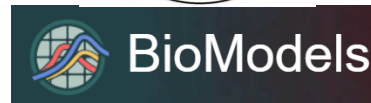
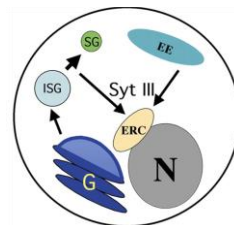
Guiding experiments



**Create Models**



libSBML



**Compile Models**

Model Reduction

System identification

**Analysis with Models**

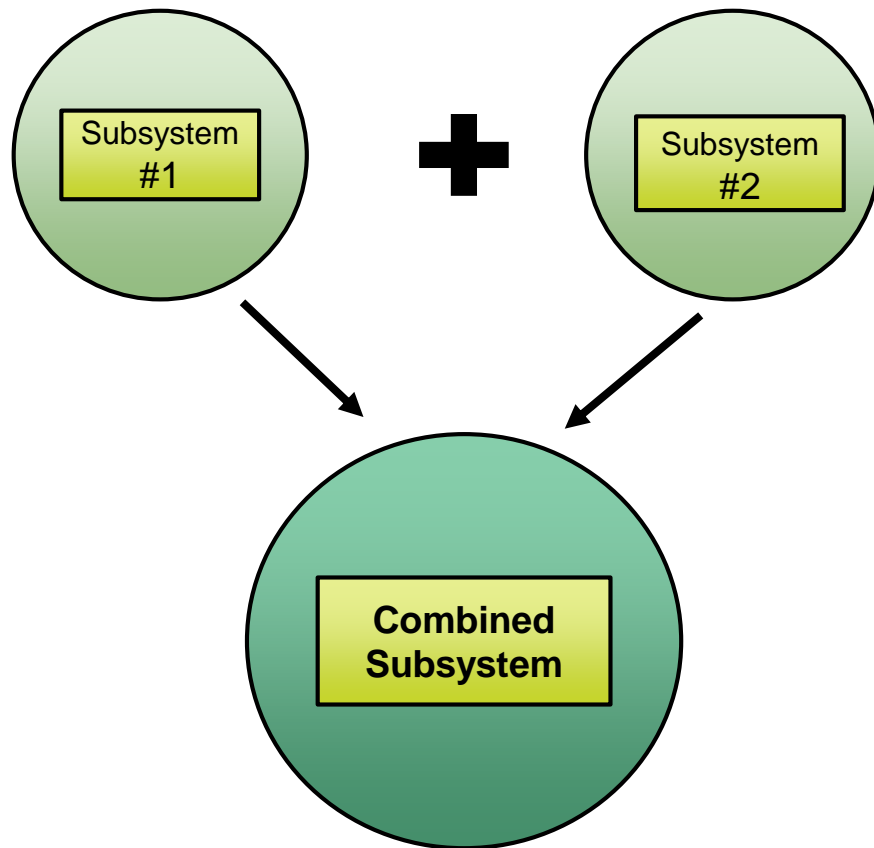
Sensitivity analysis

Guiding experiments

# Summary

# Takeaway #1

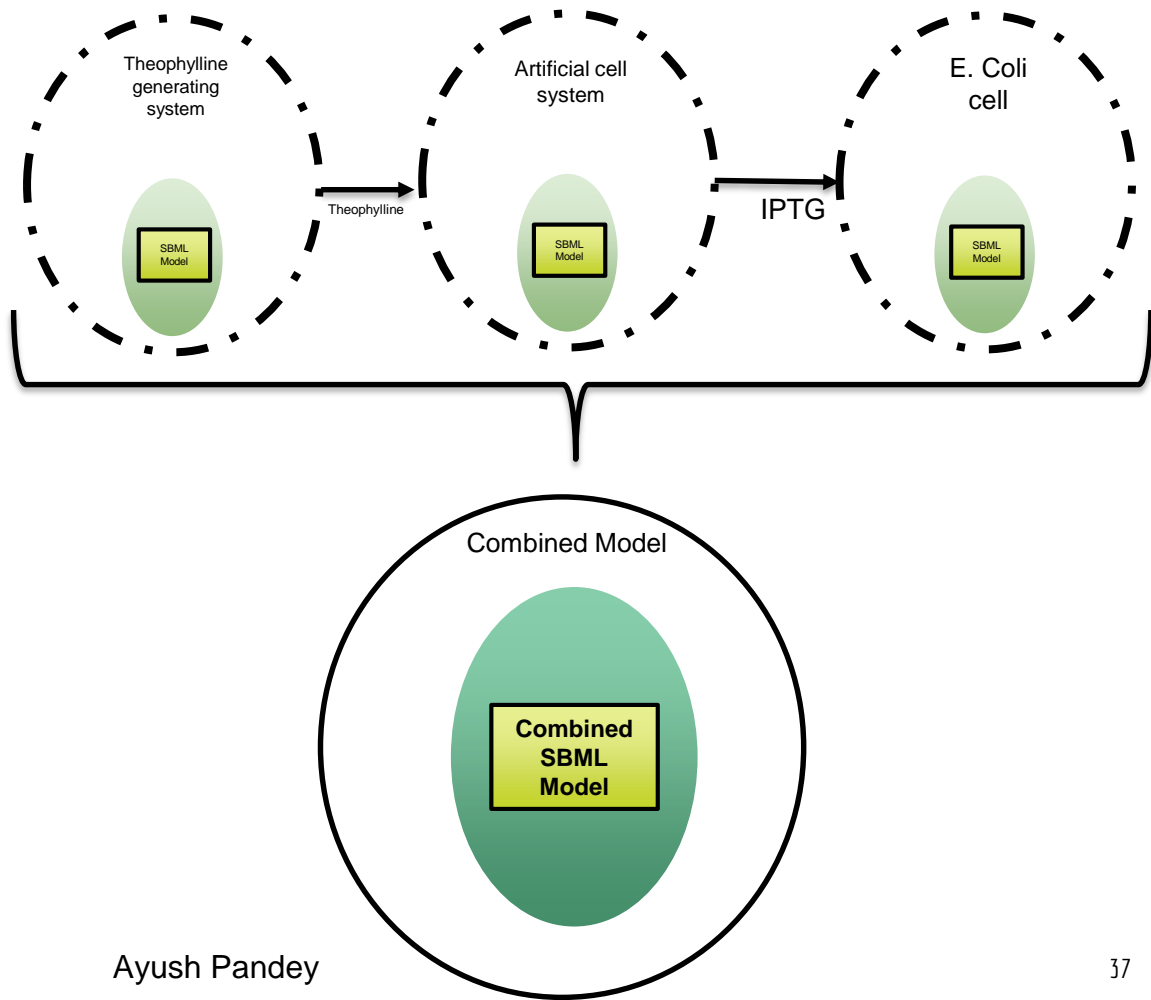
Helpful functions to  
model interactions  
between multiple SBML  
models developed using  
a “subsystem”  
framework



# Takeaway #2

Structured, and simple modeling of transport across membranes and other interactions between systems.

Important tools for model reduction can be built on this framework.



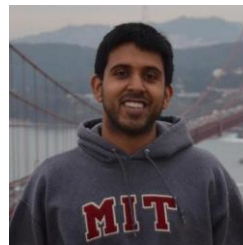
# Acknowledgments

Thanks to help and support from:

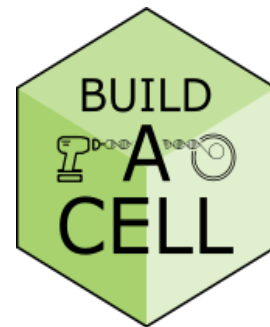
- Prof. Richard Murray



- Vipul Singhal, Anandh Swaminathan



- William Poole, Zoila Jurado



All code available and open source on –  
Github -> BuildACell -> subsbml

<https://github.com/BuildACell/subsbml/>

# Backup slides