Using R as a glue for land use and caribou

Mesachie

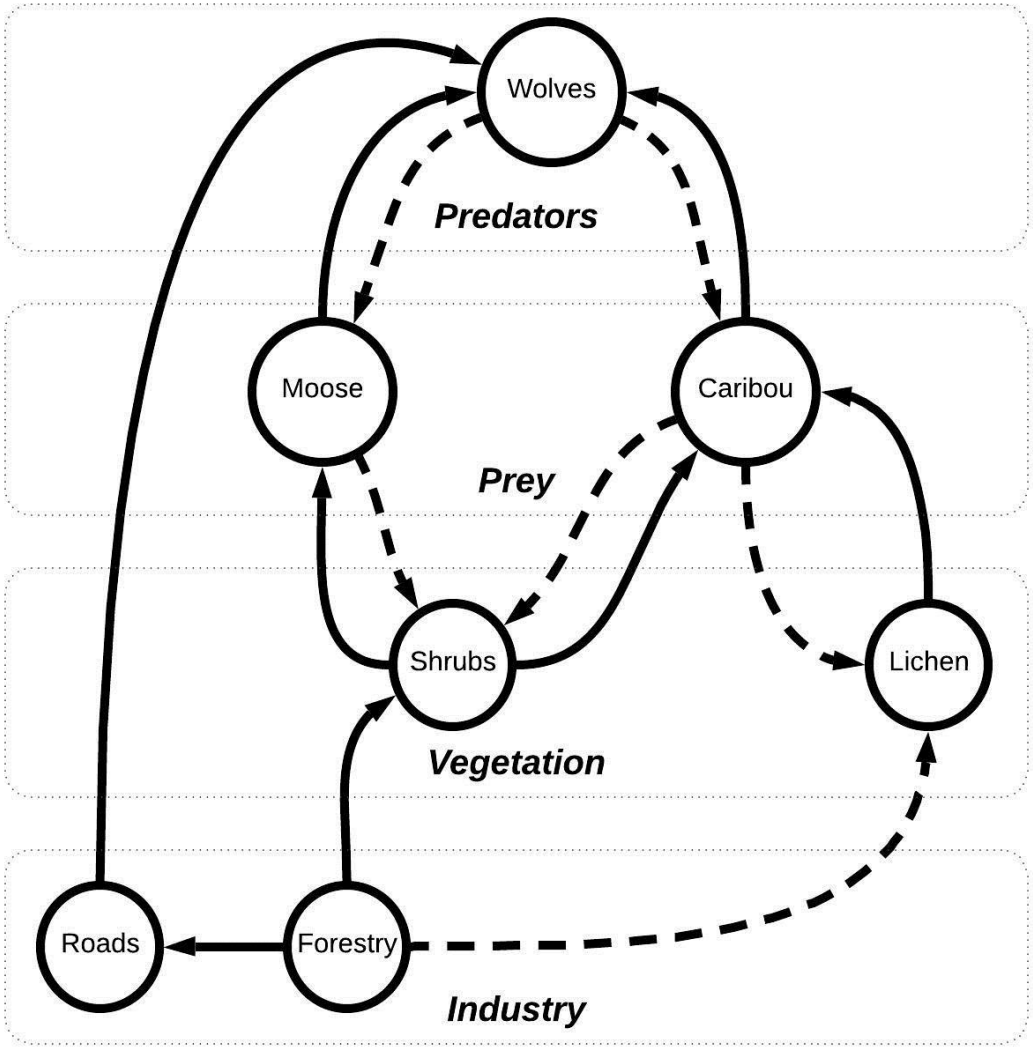
June 4, 2019

Kyle Lochhead

Outline

- Motivation
- Objectives
- Why R?
- SpaDES
- Modules

Motivation


Wolves
Predators
Moose
Caribou
Prey
Shrubs
Lichen
Vegetation
Roads
Forestry
Industry

Motivation

- Scale issue
  - Herds cross different administrative boundaries
  - Matrix habitat of adjacent management units
- Different disturbances impact caribou differently
  - Roads vs cutblocks vs fire
  - Incompatibility between disturbance thresholds and policies e.g., 500 m buffer? Partial cuts?
- Accounting for cumulative land-use impacts

Caribou and Land-Use Simulator (CLUS)

- Built in R using SpaDES
- Use(s)
  - Simulate historical and **future impacts** of land uses on indicators of the caribou-land-use system
  - **Compare** proposed policy scenarios at a range of scales and study areas
  - Provide a **transparent** model structure for rapid feedback when exploring the decision space
  - **Communicate** decision spaces for any herd across the province

Why R?

Pros

- Free
- Well supported – comprehensive library
- Documentation
- Post hoc – communication piece
  - Visuals and reporting
- Flexible for supporting many languages

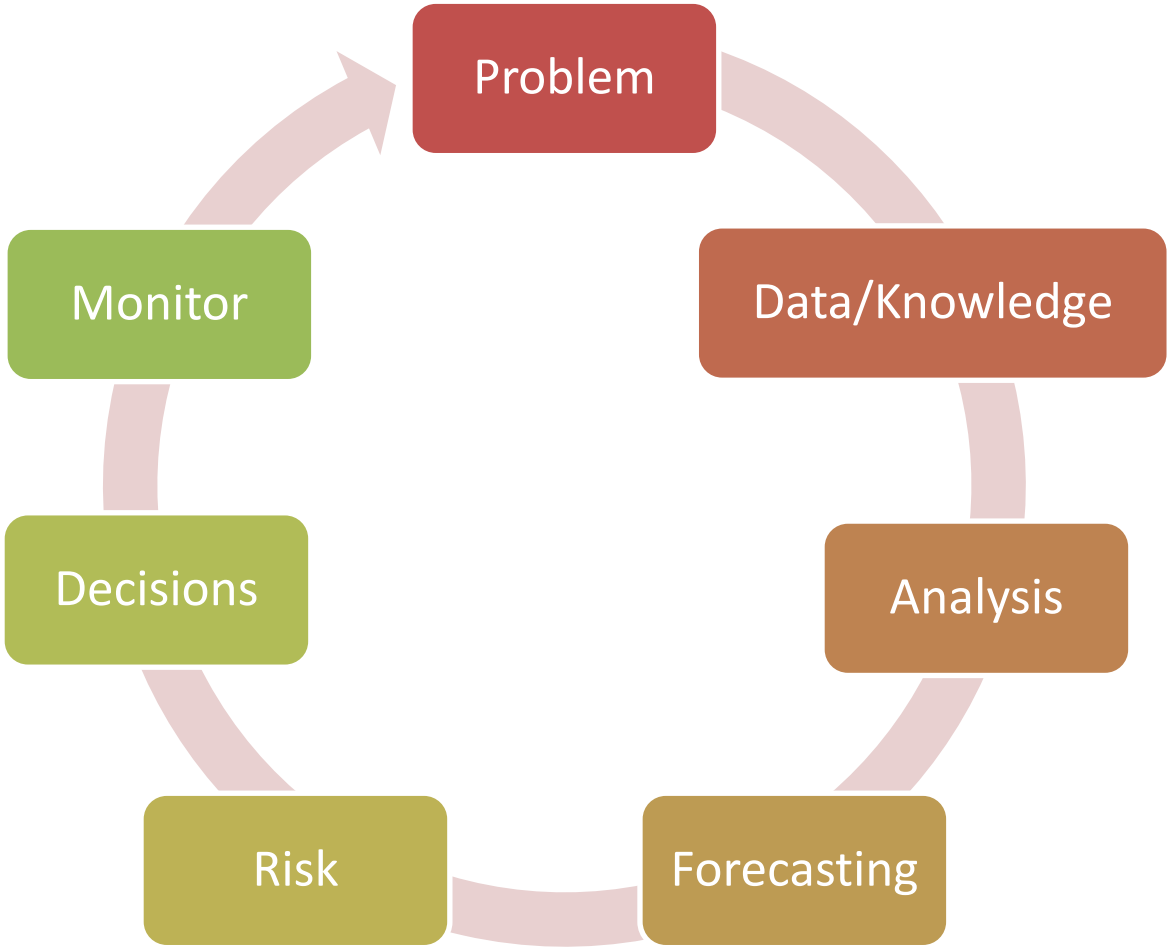
Cons

- “Its slow…” – default implementation is interpreted
- Not all packages are “useful” – takes time to dive into what is going on

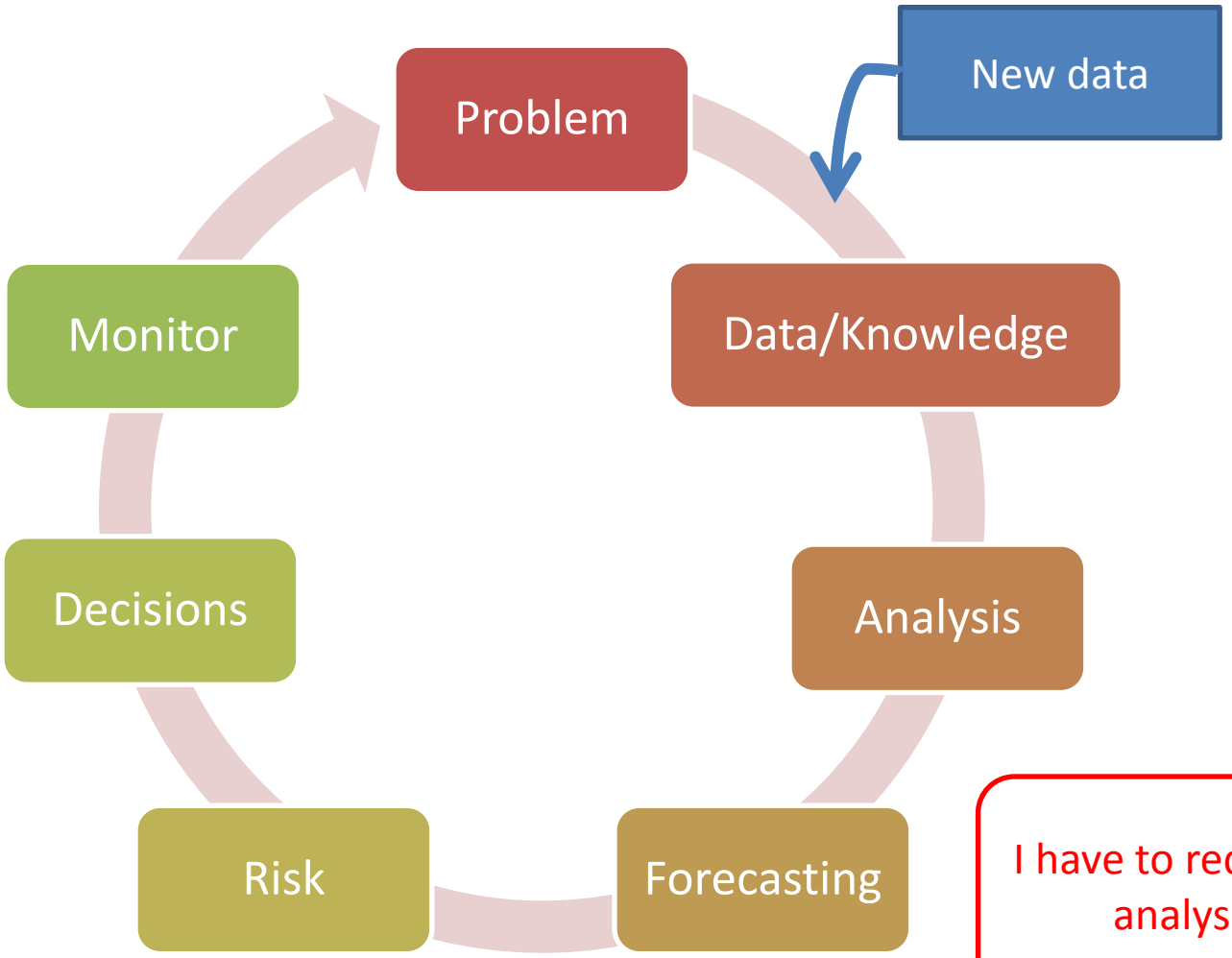
SpaDES an R package

- Developed by NRCan
- Spatial Discrete Event-based Simulation
- Attempts to overcome -> many models, low integration
- Principles
  - Transparency
  - Visualization
  - Reproducibility
  - Modularity
  - Scalability
  - Traceability
  - Sensitivity

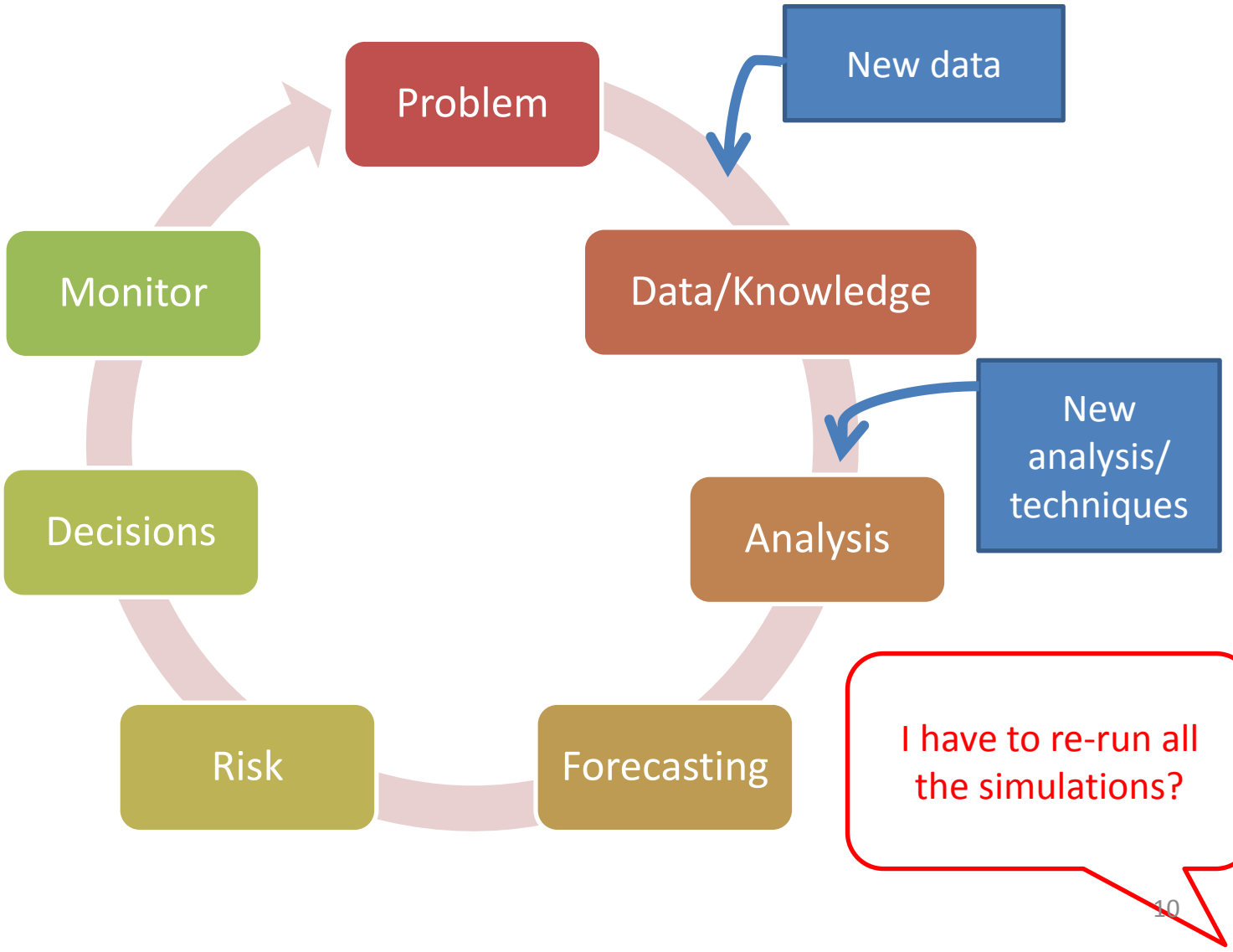
What does this mean?



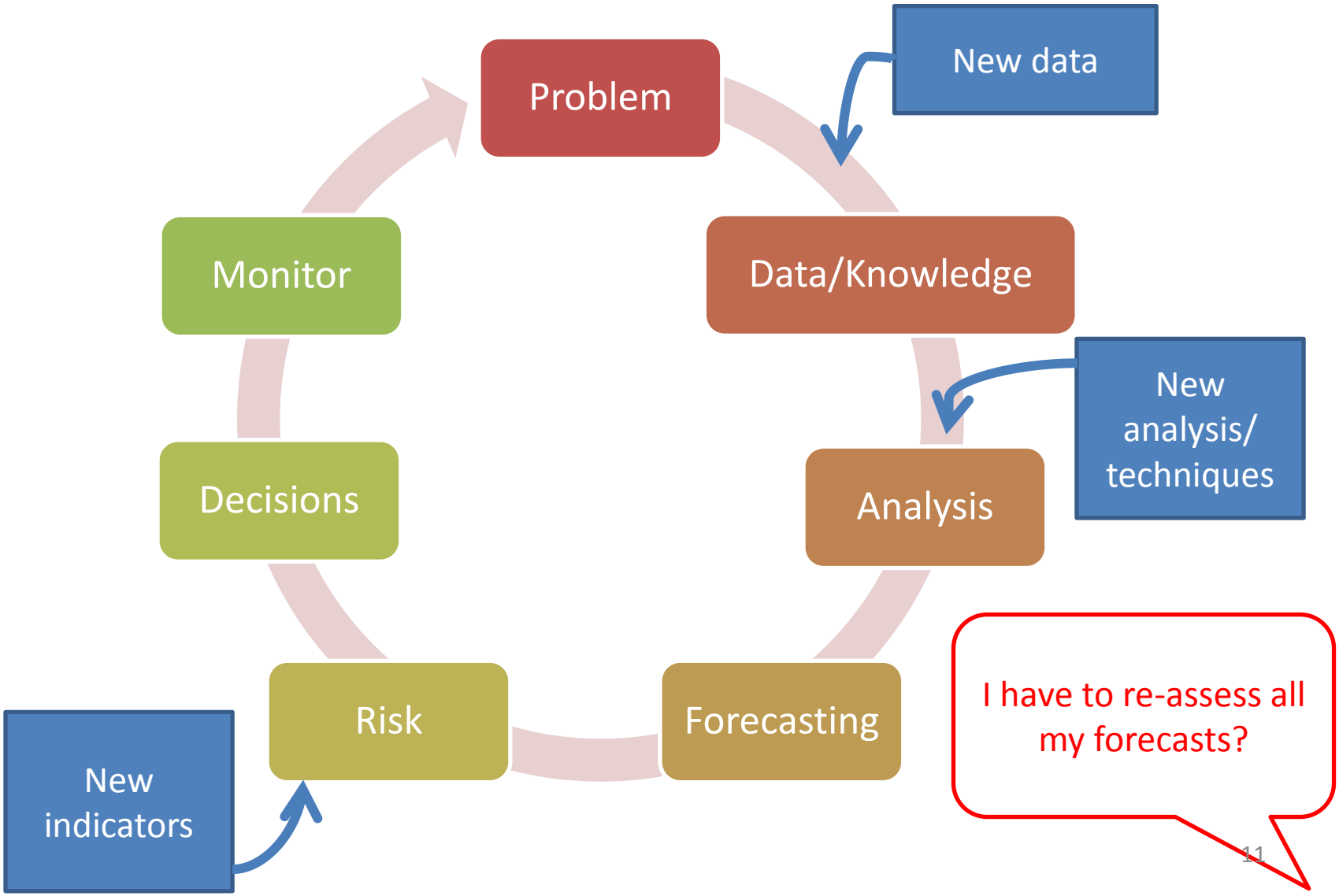
What does this mean?



What does this mean?

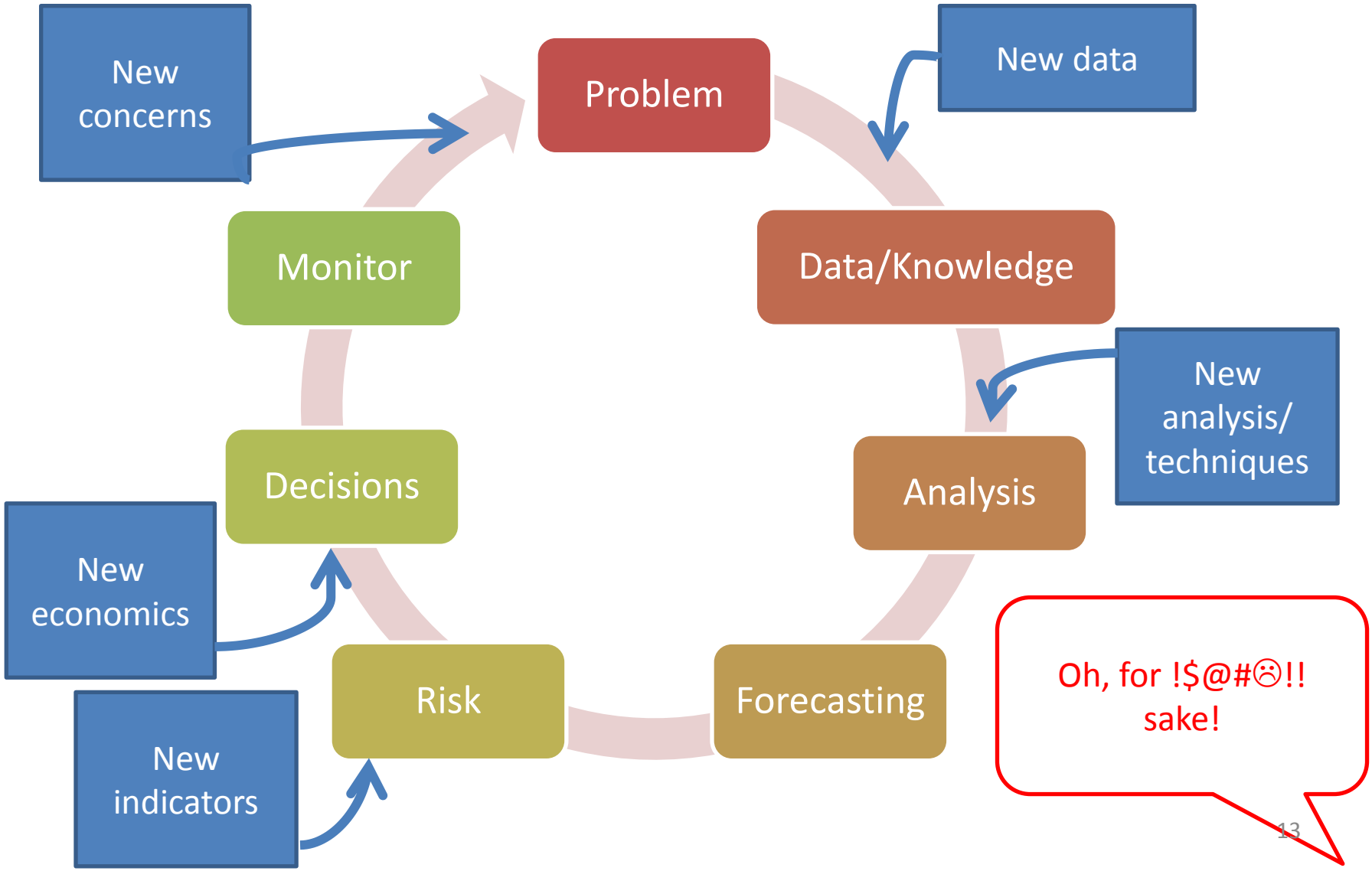


What does this mean?



What does this mean?

What does this mean?



CLUS Concept

- Build spatial scenarios in real time via Scenario Tool
- Build a SQLite database - clusdb
- Use R to traverse: data manipulation, linkages to other languages and reporting/visualization

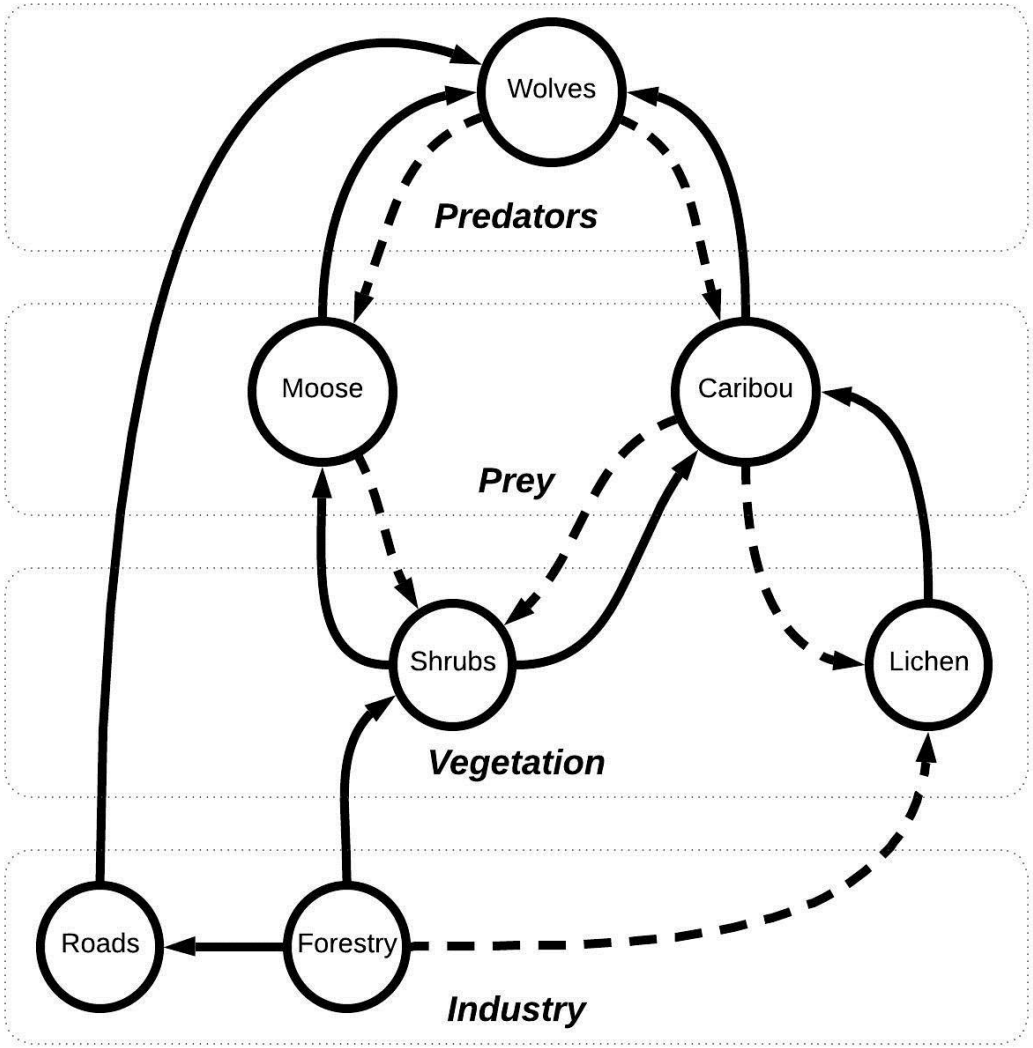
CLUS Concept

- **Transparent**, all data and information are in a structured - designed database. All code freely available. Adding new or improving old is encouraged and easy via SpaDES + github
- **Reproducible**, connects to Postgresql which supports back and forth between vector and raster. Reproducible package supports version control of cached outputs.
- **Modular**, each process is itself a “model”. Connect to other “modules” for insects, fire, growth, birds, wolves, moose, etc
- **Scalable**, SQL is optimized to handle large queries. Leverages data.table package on R-side (millions of records in seconds)
- **Traceability**, basic error/warning reporting, can save/print out any line of code, procedural R code
- **Sensitivity**, core SpaDES was developed to handle caching of many runs. I implement parallel instances across multi-workstations for some modules (i.e., blocking). Spades.core::experiment() – stores many replications of stochastic components of models

Modules

- Data Prep (dataloaderCLUS)
  - leverages PostgreSQL
- Growth and yield (growingStockCLUS)
  - leverages SQLite
- Resource Selection Function (rsfCLUS)
  - leverages R
  - Caribou habitat selection model (Tyler)
- Roads (roadsCLUS)
  - leverages C++
  - “snap”, “lcp”, “mst”
- Blocking (blockingCLUS)
  - leverages Java
  - “pre-solve”, “dynamic”
- Harvesting (forestryCLUS)
  - leverages SQLite
  - Spatial harvest simulator. S.T. various management constraints
  - Harvest queue based on a simple priority
    - Future –add optimization (Q3)

Motivation


Wolves
Predators
Moose
Caribou
Prey
Shrubs
Lichen
Vegetation
Roads
Forestry
Industry