Welcome!

FEW: A Sustainable Rural Framework for the Upper Great Plains

Twitter: #SDMinesFEW

Thank you to our sponsors:

Goals of the Workshop

1. Exchange new ideas for FEW
2. Cross-pollinate → establish new and strengthen existing collaborations amongst our groups.
3. Prepare teams in advance to pursue new NSF and other (DOE, USDA, USAID, and other) FEW-related funding opportunities
4. Provide guidance (white paper) to NSF for future FEW funding initiatives (INFEWS, many others)

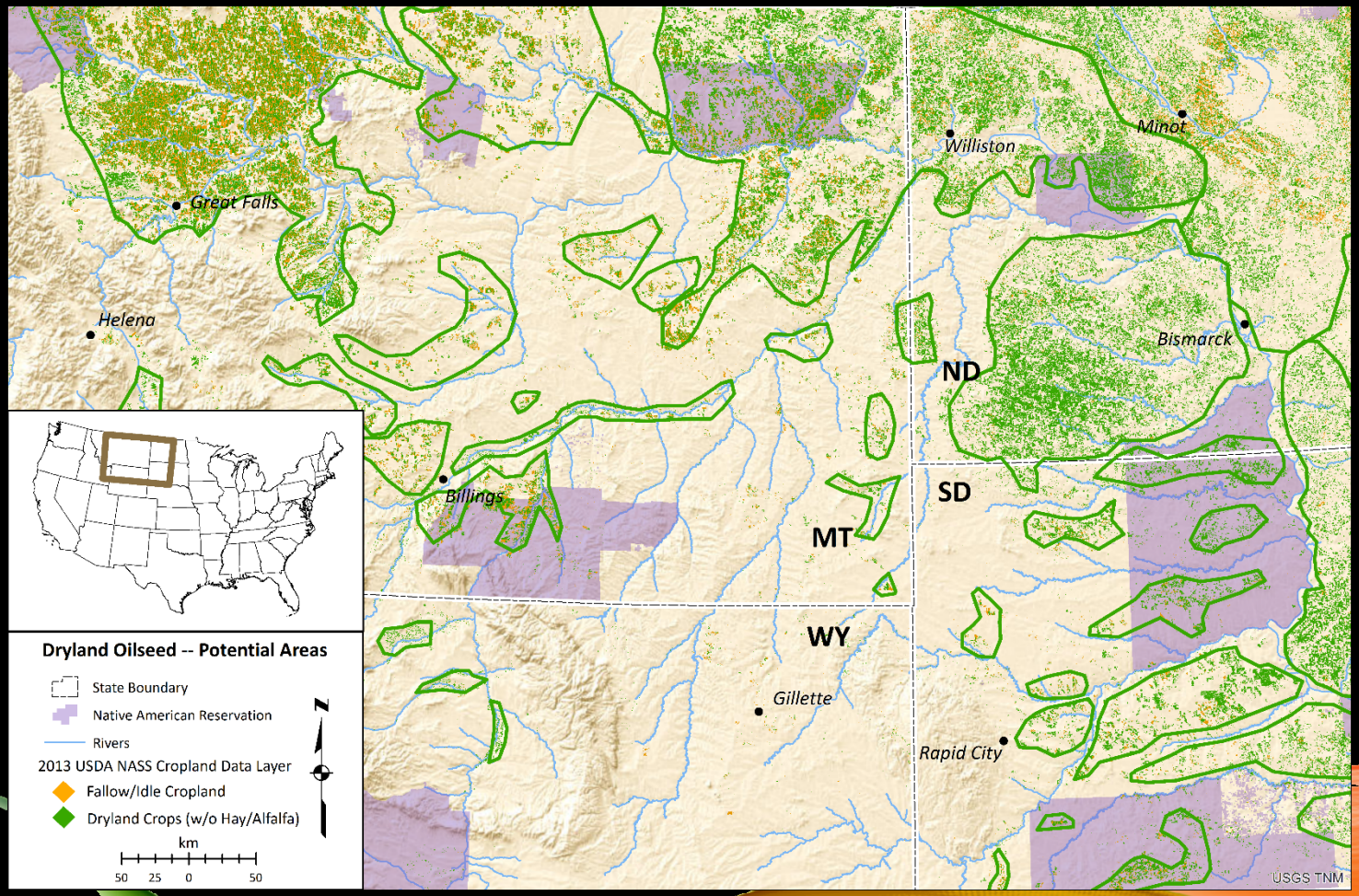
Plan for Tuesday – Day 2

- **Identifying fundamental problems**
- **Food session + panel discussions**
  MIT, Argonne NL, NDSU
- **Nexus session + panel discussions**
  SDSU, Oak Ridge NL
- **White paper breakout groups**
  Focusing on NSF recommendations and team building

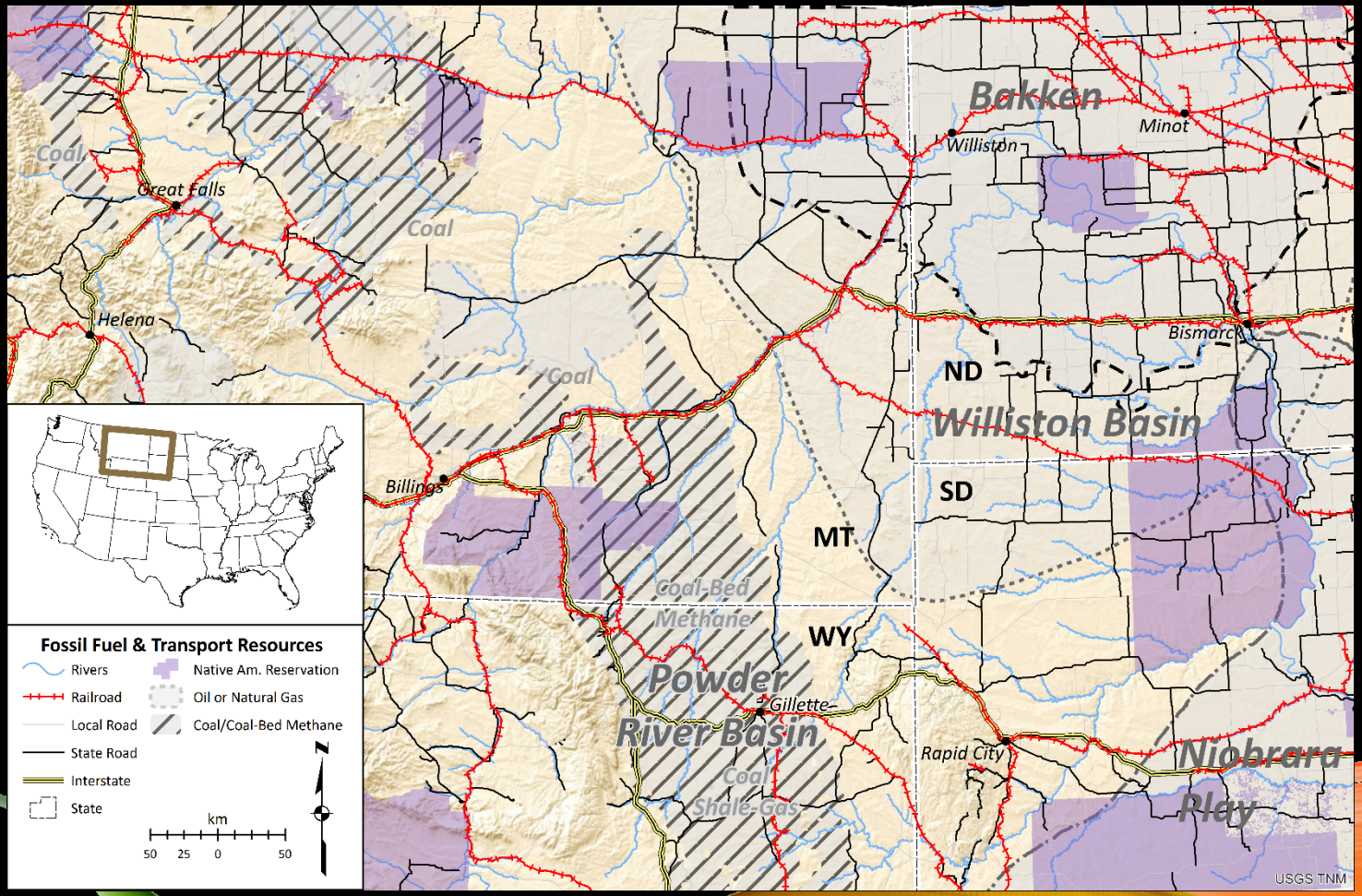
Suggestions

1. For NSF – think big! Look at what you can't do, and seek out people who can
2. What are the FEW game changers. No magic pill exists (that I'm aware of).
3. Think about the challenges and tools needed to create, solve, attack FEW
4. Network

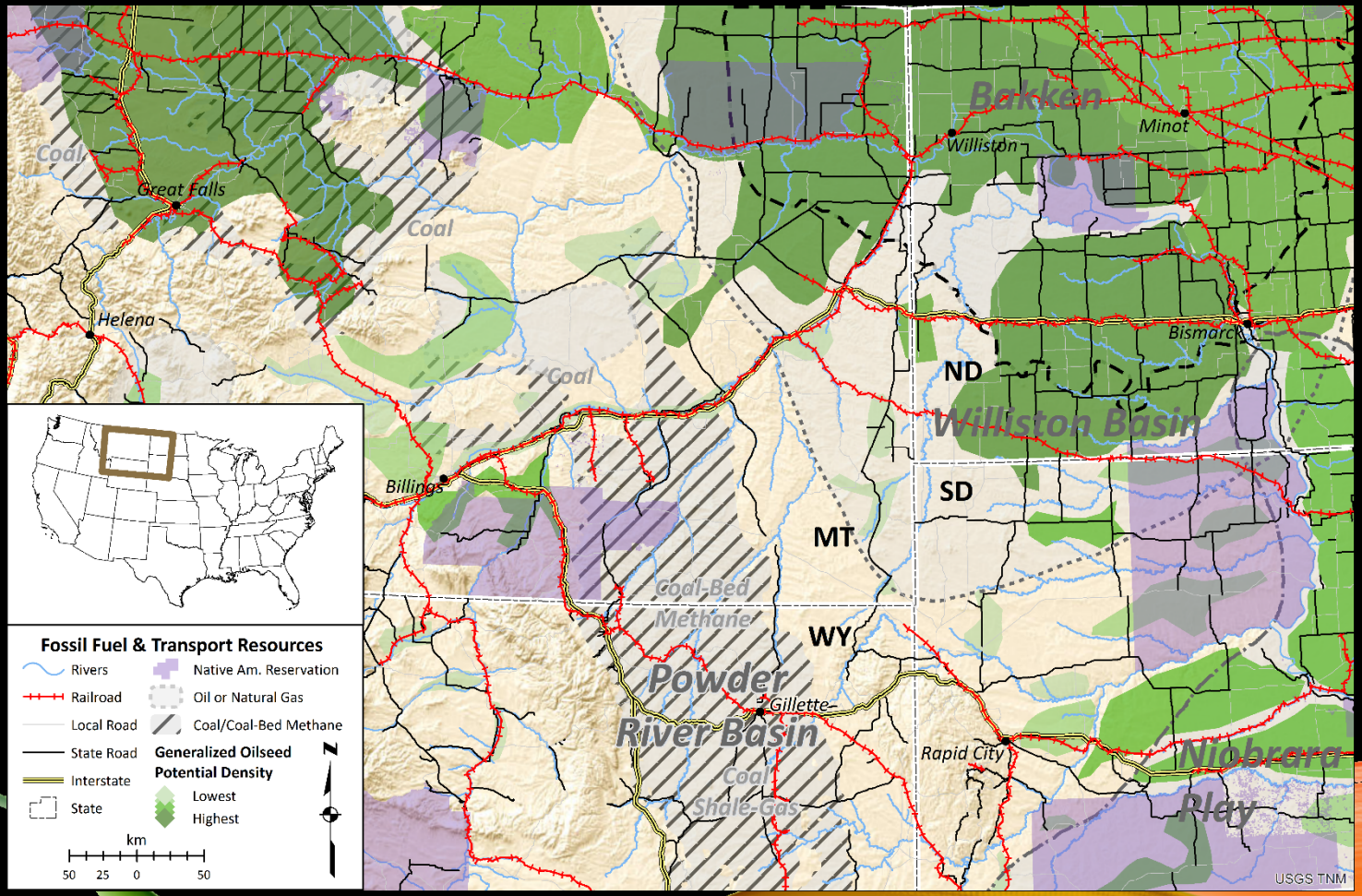
NON-FOOD OILSEEDS POTENTIAL



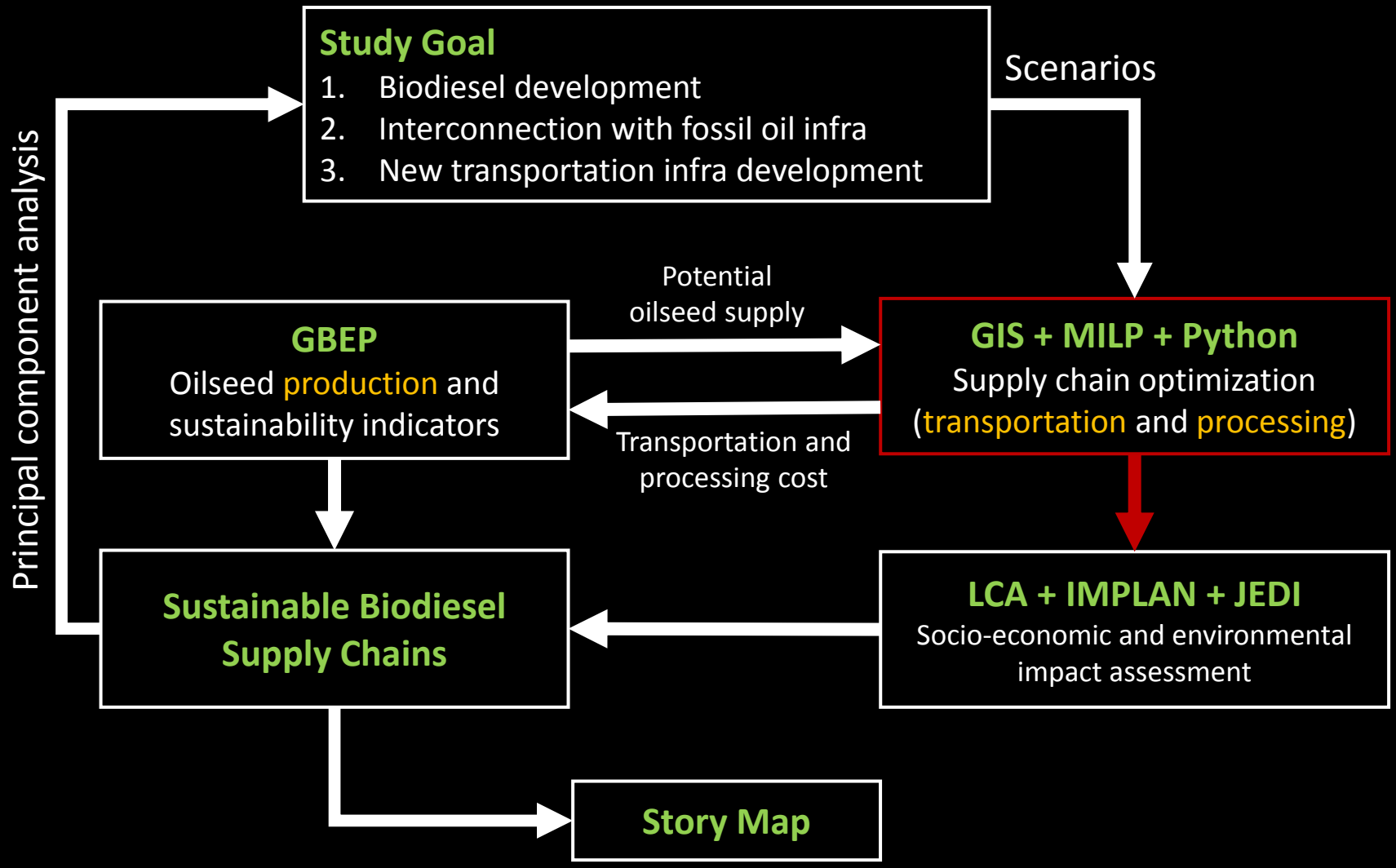
FOSSIL RESOURCES



FOSSIL RESOURCES



Study Approach



Supply Chain Optimization Process

```
43  arc1Data = {}
44  with open ('C:\\GIS\\arc1.csv') as csvfile:
45      reader = csv.DictReader(csvfile)
46      for row in reader:
47          arc1Data[row['elvtrID'], row['rfnrID']] = row['totalMiles']
48          Arcs1 = [(k) for k in arc1Data.keys()]
49
50  arc2Data = {}
51  with open ('C:\\GIS\\arc2.csv') as csvfile:
52      reader = csv.DictReader(csvfile)
53      for row in reader:
54          arc2Data[row['rfnrID'], row['ctID']] = row['totalMiles']
55          Arcs2 = [(k) for k in arc2Data.keys()]
56
57  # Splits the dictionaries to be more understandable
58  (bU, bS) = splitDict(rfnrData)
59
60  # Creates the boundless Variables as Continuous
61  x1 = LpVariable.dicts("Route1",Arcs1,0,None,LpContinuous)
62  x2 = LpVariable.dicts("Route2",Arcs2,0,None,LpContinuous)
63
64  # Creates the 'prob' variable to contain the problem data
65  prob = LpProblem("Biodiesel meal assignment",LpMinimize)
66
67  # Creates the objective function   ***DOES NOT HAVE THE VARIABLE COS
68  prob +=  lpSum([int(arc1Data[i])*0.0015*x1[i] for i in Arcs1]) + lpS
69           "Total Cost of Transport"
70
71  # Creates all problem constraints - this ensures the amount going in
72  for n in Refineries:
73      prob += int(bS[n])*0.001*lpSum([x1[(i,j)] for (i,j) in Arcs1 if
74  for n in Dieseldmd:
75      prob += lpSum([x2[(i,j)] for (i,j) in Arcs2 if j == n]) == int(c
76  for n in Elevators:
77      prob += lpSum([x1[(i,j)] for (i,j) in Arcs1 if i == n]) <= int(e
78  for n in Refineries:
79      prob += lpSum([x2[(i,j)] for (i,j) in Arcs2 if i == n]) <= int(b
80
81  # The problem data is written to an .lp file
82  prob.writeLP("BasicScenario.lp")
83
84  # The problem is solved using PuLP's choice of Solver
85  prob.solve()
86
87  # The status of the solution is printed to the screen
88  print "Status:", LpStatus[prob.status]
89
```

Interactive Window

```
Status: Optimal
>>> Route1_('1',_'1') = 0.0
Route1_('1',_'10') = 0.0
Route1_('1',_'11') = 0.0
Route1_('1',_'12') = 0.0
Route1_('1',_'13') = 0.0
Route1_('1',_'14') = 0.0
Route1_('1',_'15') = 0.0
Route1_('1',_'16') = 0.0
Route1_('1',_'17') = 0.0
Route1_('1',_'2') = 14068595.0
Route1_('1',_'3') = 0.0
Route1_('1',_'4') = 0.0
Route1_('1',_'5') = 0.0
Route1_('1',_'6') = 0.0
```

```
Route2_('8',_'1') = 0.0
Route2_('8',_'2') = 0.0
Route2_('8',_'3') = 0.0
Route2_('8',_'4') = 0.0
Route2_('8',_'5') = 0.0
Route2_('8',_'6') = 0.0
Route2_('9',_'1') = 0.0
Route2_('9',_'2') = 0.0
Route2_('9',_'3') = 0.0
Route2_('9',_'4') = 0.0
Route2_('9',_'5') = 0.0
Route2_('9',_'6') = 0.0
Total Cost of Transportation =  68003838.7855
```

Breakout Questions

1. What would be ideal 'goals' to meet FEW sustainability for our region?
2. What considerations should be included for NSF?
3. What are the tools/data/technologies needed?
4. What behavior changes are needed?
5. What are the components of a successfully funded proposal? Who are the players and their roles?