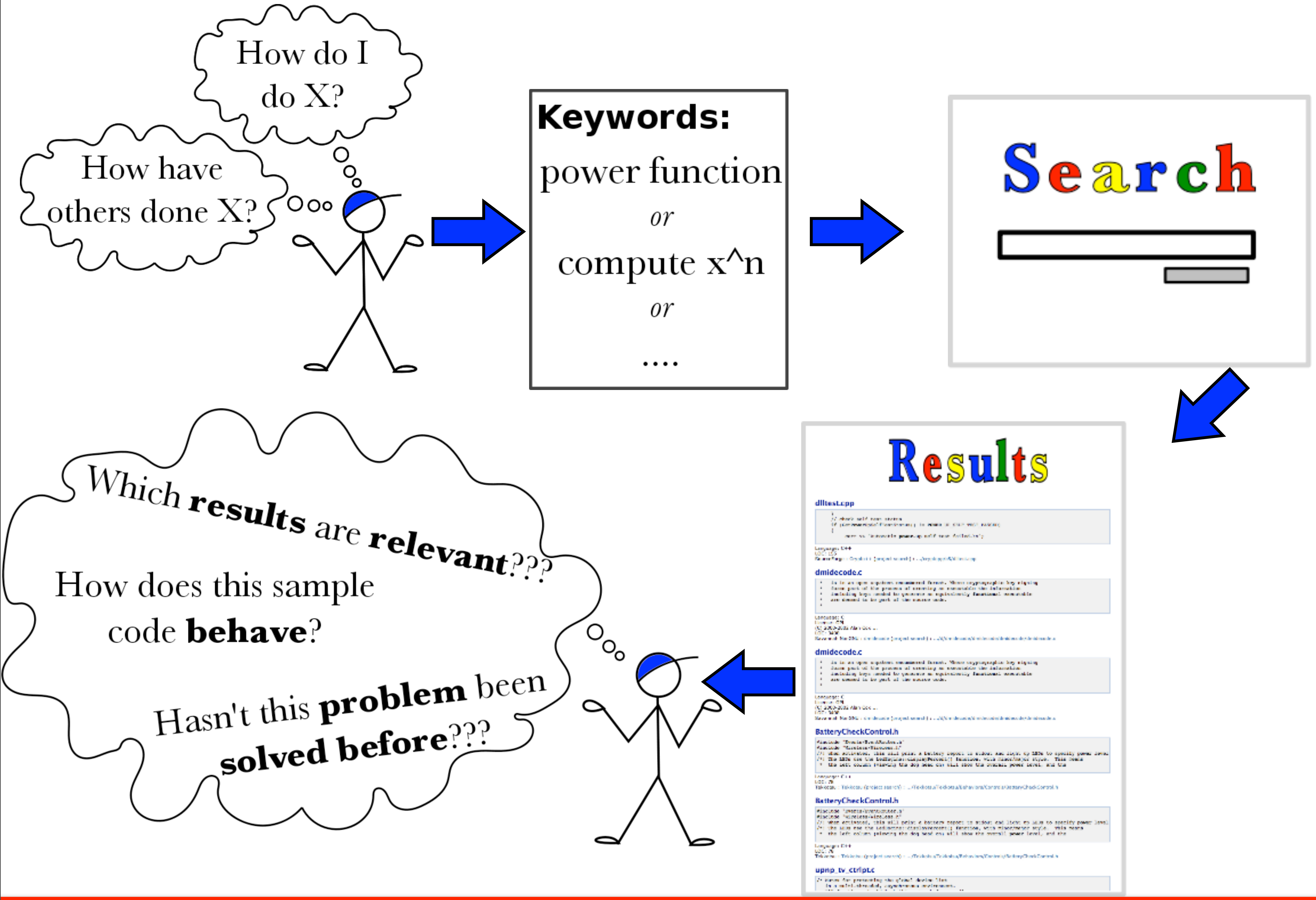
State of the Practice Code Search



How do I do X?

How have others done X?

Keywords:
power function
or
compute x^n
or
....

Search

Results

dlltest.cpp

dmidecode.c

dmidecode.c

BatteryCheckControl.h

BatteryCheckControl.h

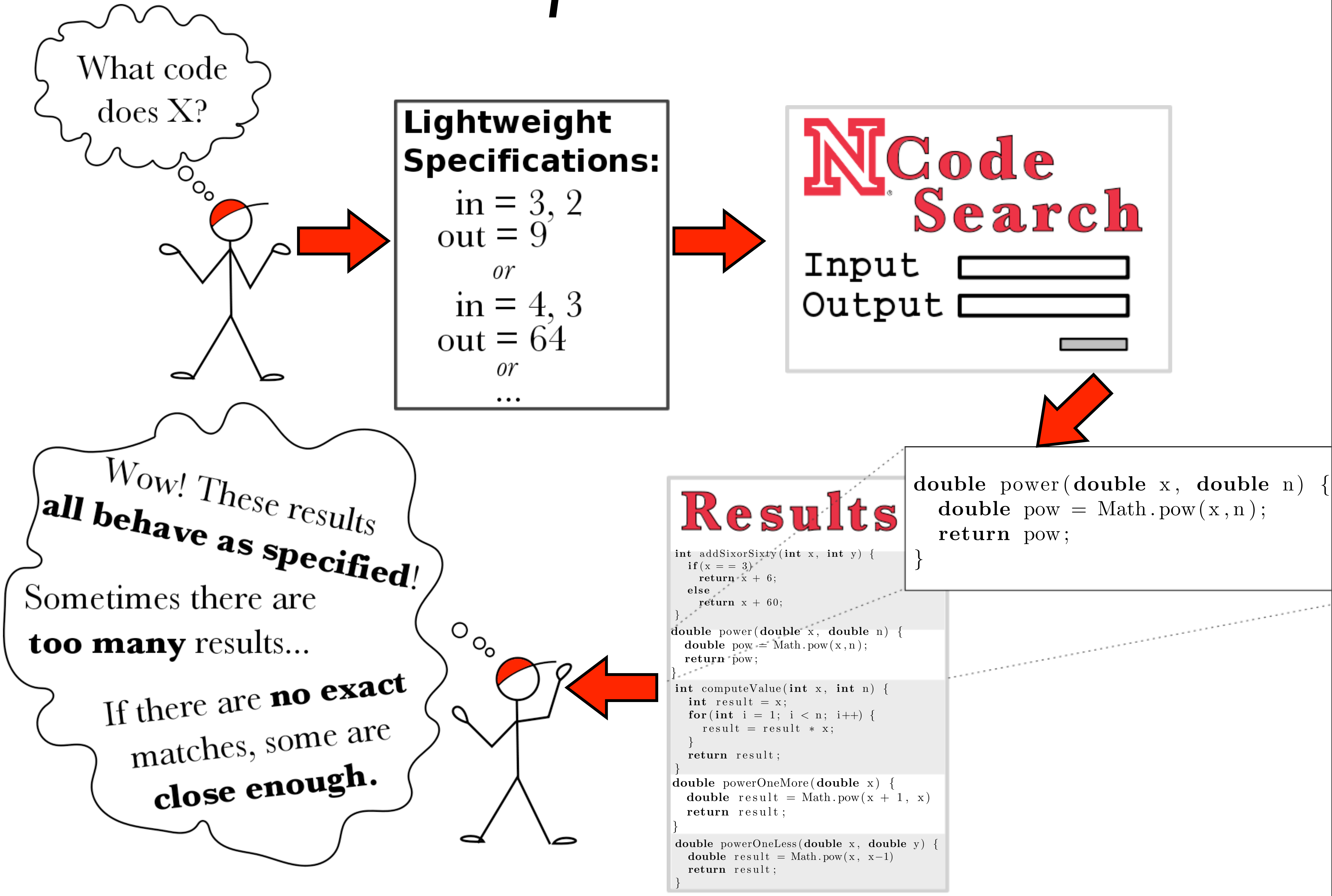
upnp_tv_ctrlpt.c

Which **results** are **relevant**???

How does this sample code **behave**?

Hasn't this **problem** been **solved before**???

Proposed Code Search


What code does X?
Wow! These results **all behave as specified!**
Sometimes there are **too many** results...
If there are **no exact** matches, some are **close enough.**

Lightweight Specifications:

in = 3, 2
out = 9
or
in = 4, 3
out = 64
or
...

NCode Search

Input
Output

Results

```
int addSixorSixty(int x, int y) {
  if(x = = 3)
    return x + 6;
  else
    return x + 60;
}
double power(double x, double n) {
  double pow = Math.pow(x,n);
  return pow;
}
int computeValue(int x, int n) {
  int result = x;
  for(int i = 1; i < n; i++) {
    result = result * x;
  }
  return result;
}
double powerOneMore(double x) {
  double result = Math.pow(x + 1, x)
  return result;
}
double powerOneLess(double x, double y) {
  double result = Math.pow(x, x-1)
  return result;
}
```

```
double power(double x, double n) {
  double pow = Math.pow(x,n);
  return pow;
}
```

Research Contribution

- An approach for **semantic search** via **lightweight specs**
- Uses an SMT solver to **solve the search**
- **Promotes reuse** of repository code
- Benefits over state-of-the-art semantic search:
  - ✓ **Cost of Query**: I/O is easier to write than formal specifications
  - ✓ **Cost of Search**: candidate code is not executed
- Benefits over state-of-the-practice syntactic search:
  - ✓ **Relevance of Results**: all responses behave as specified, and close-enough matches can be identified

Challenges

- **Encoding** programs as constraints
- **Relaxing** constraints when no exact matches exist
- **Evaluating** the cost of our search vs. traditional search from two perspectives: *efficiency* and *effectiveness*