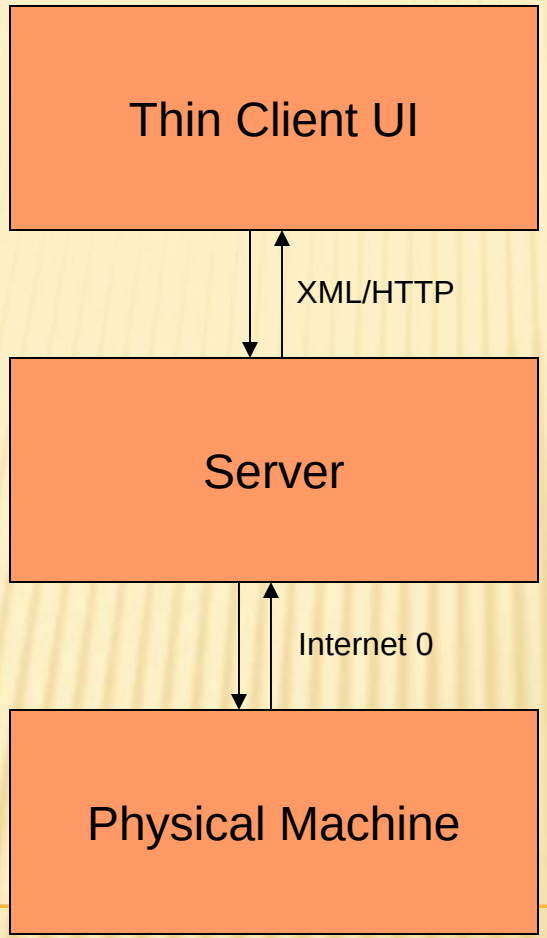
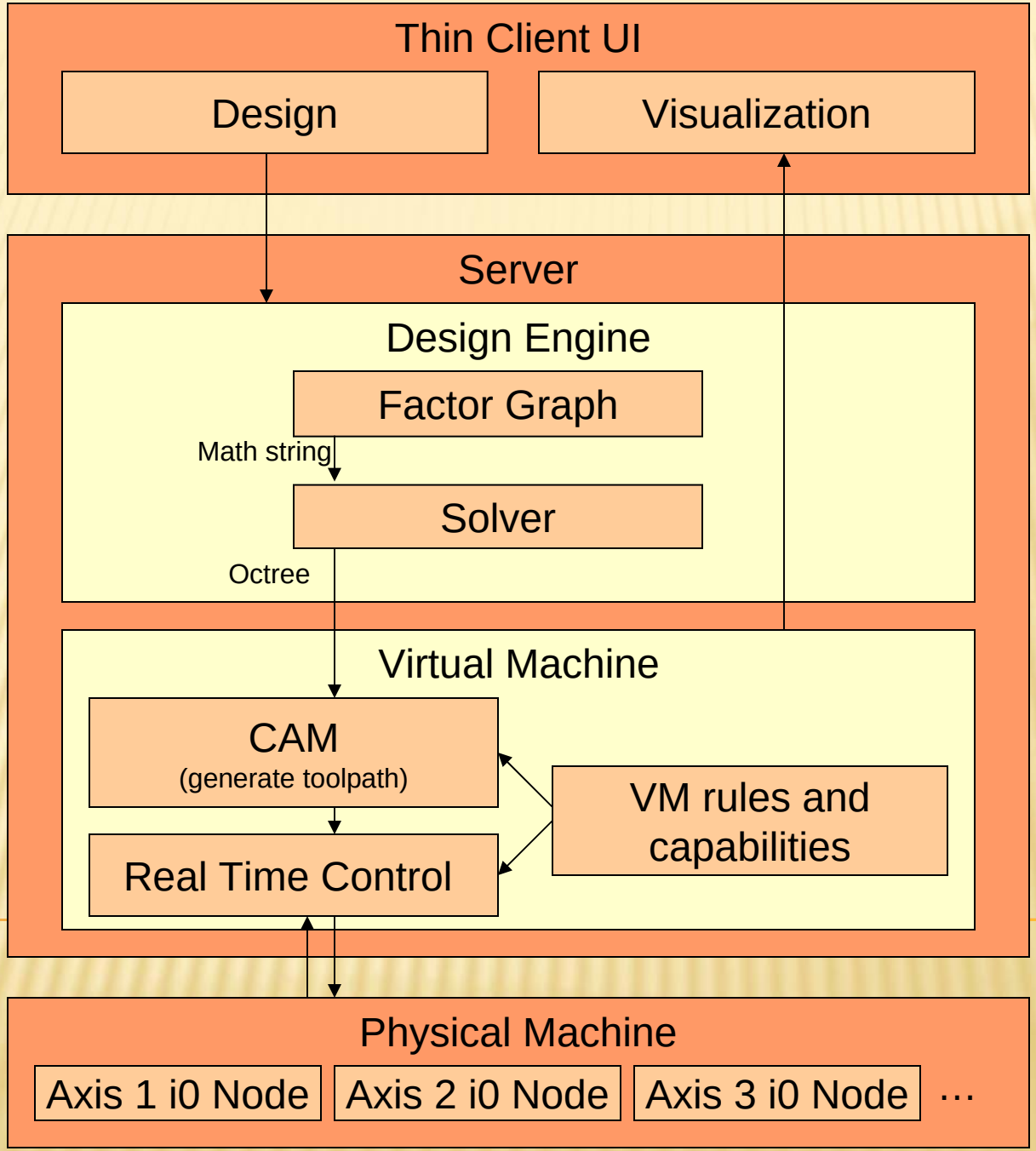
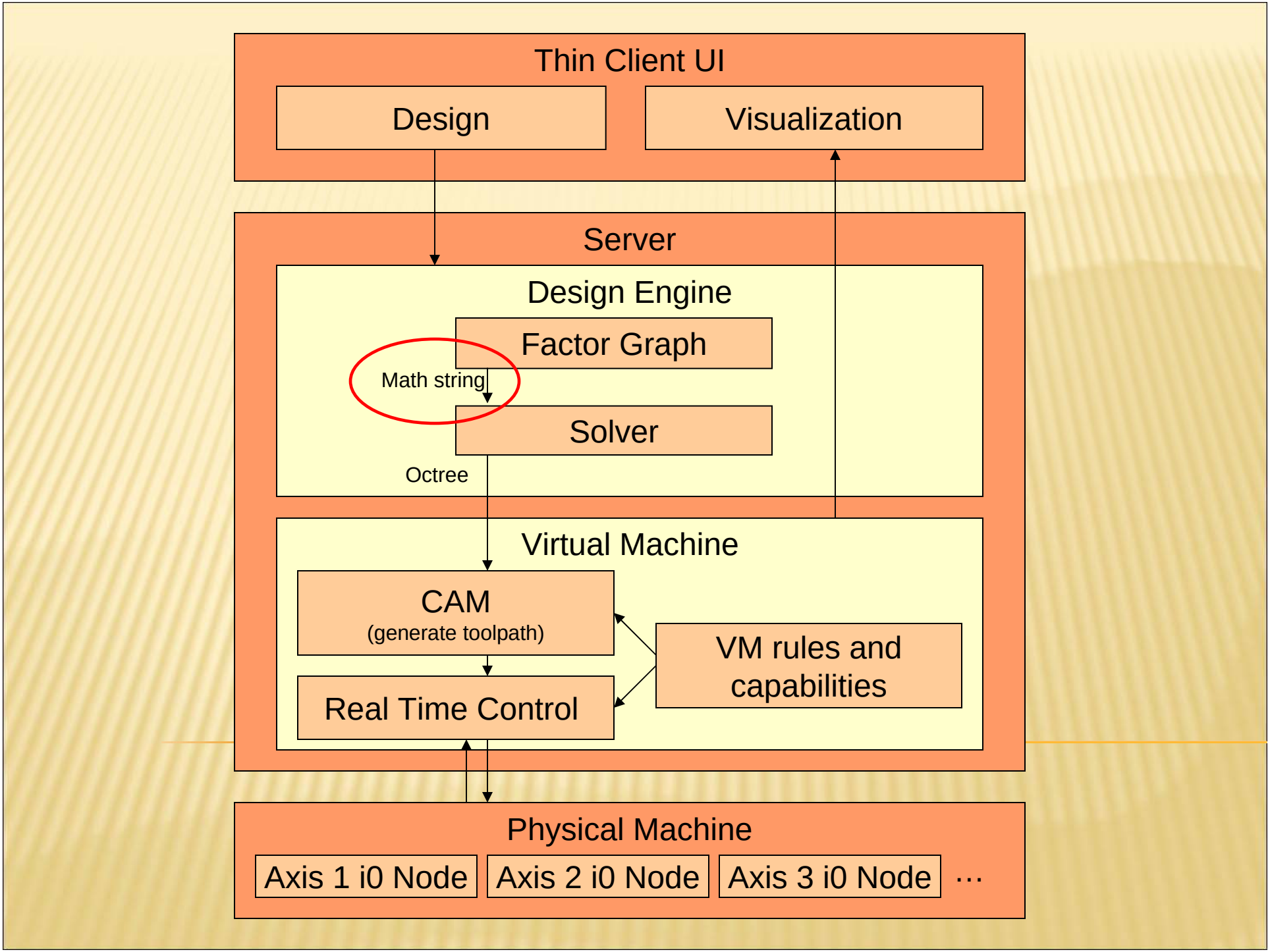
Rapid-Prototyping of Rapid-Prototyping Machines

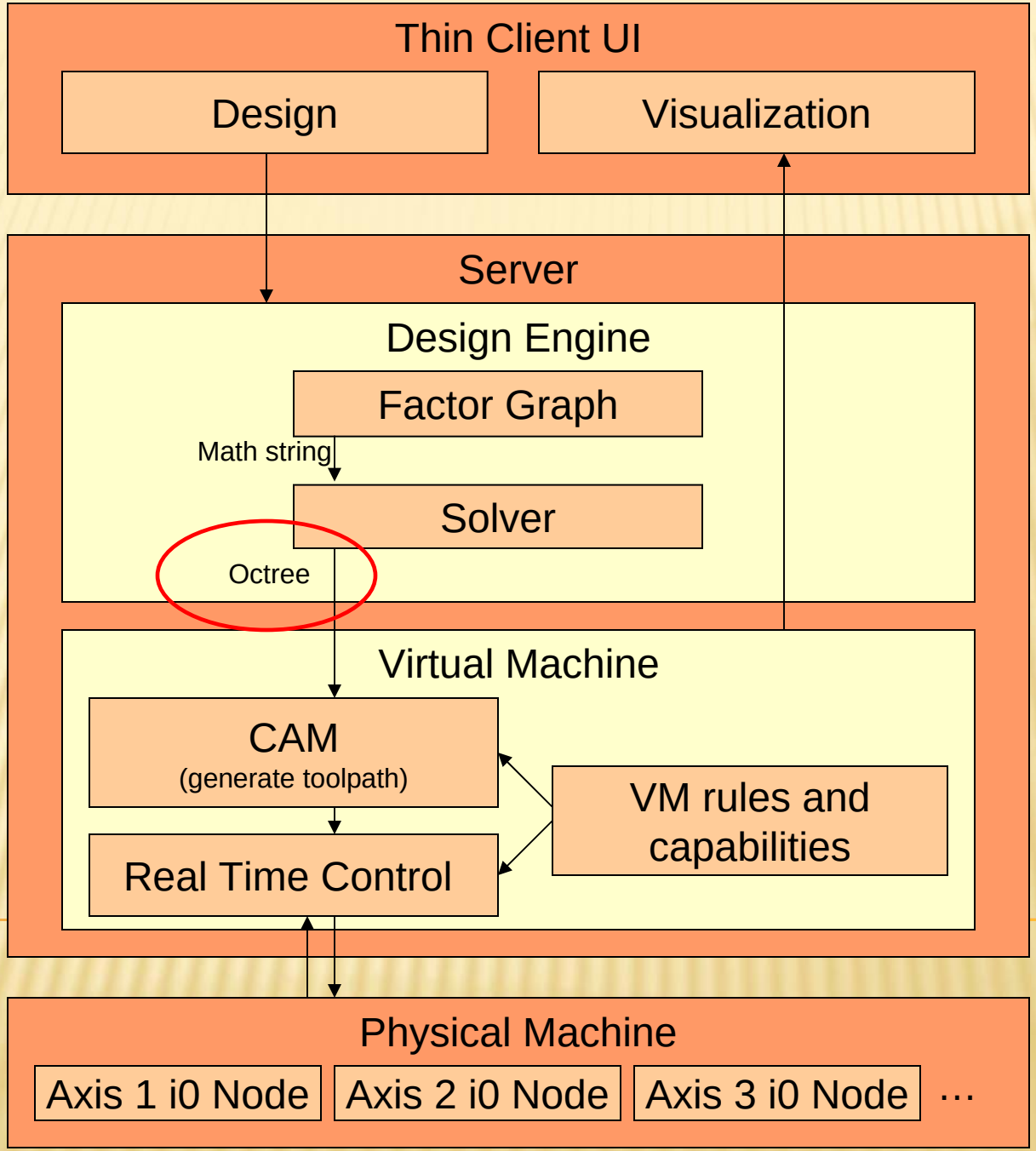
The Inner Workings of the Software

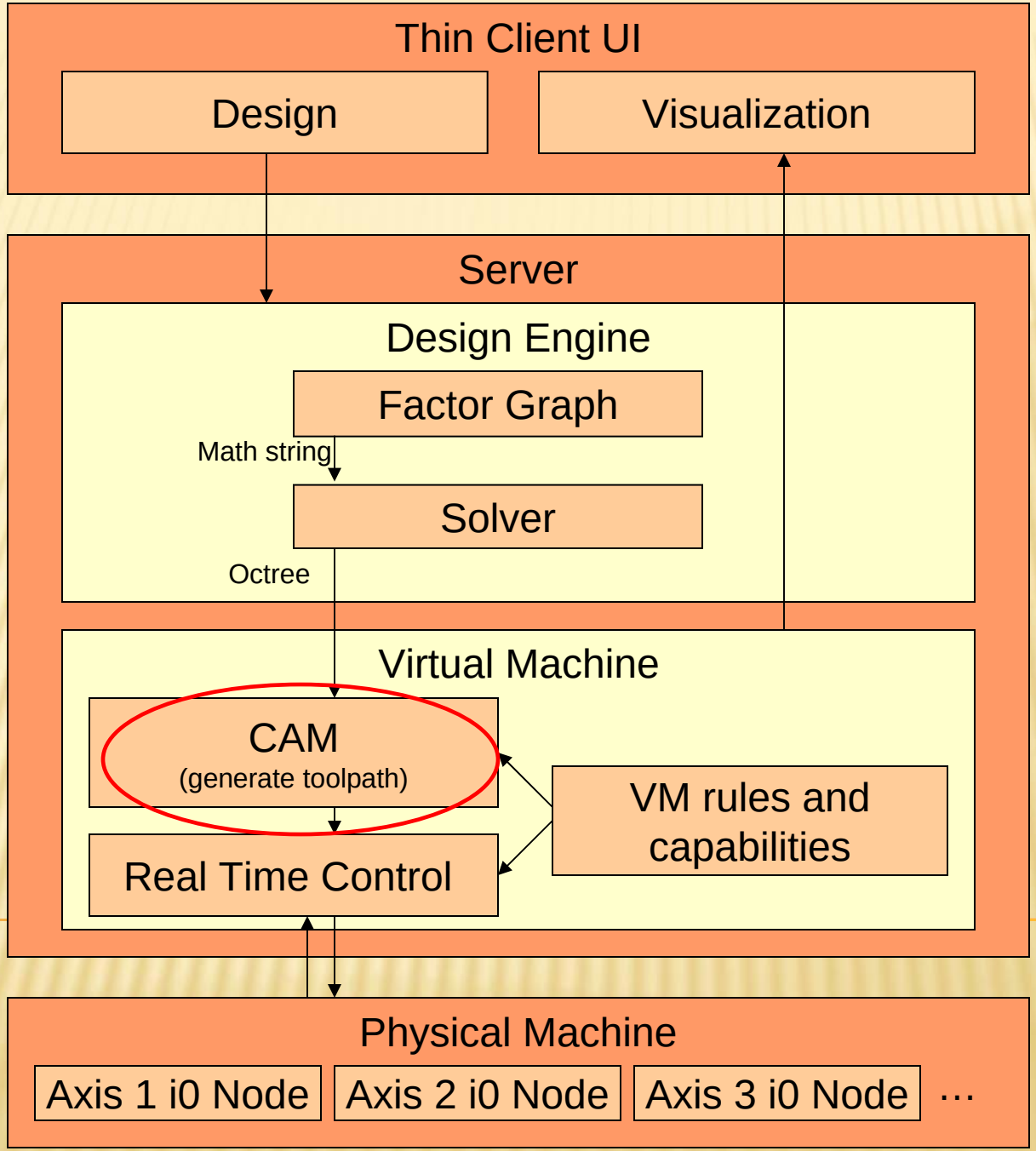
Steve Leibman
sleibman@mit.edu


Thin Client UI
XML/HTTP
Server
Internet 0
Physical Machine


Thin Client UI
Design
Visualization
Server
Design Engine
Factor Graph
Math string
Solver
Octree
Virtual Machine
CAM
(generate toolpath)
Real Time Control
VM rules and capabilities
Physical Machine
Axis 1 i0 Node
Axis 2 i0 Node
Axis 3 i0 Node
…


Thin Client UI
Design
Visualization
Server
Design Engine
Factor Graph
Math string
Solver
Octree
Virtual Machine
CAM
(generate toolpath)
Real Time Control
VM rules and capabilities
Physical Machine
Axis 1 i0 Node
Axis 2 i0 Node
Axis 3 i0 Node
…


Thin Client UI
Design
Visualization
Server
Design Engine
Factor Graph
Math string
Solver
Octree
Virtual Machine
CAM
(generate toolpath)
Real Time Control
VM rules and capabilities
Physical Machine
Axis 1 i0 Node
Axis 2 i0 Node
Axis 3 i0 Node
…


Thin Client UI
Design
Visualization
Server
Design Engine
Factor Graph
Math string
Solver
Octree
Virtual Machine
CAM
(generate toolpath)
VM rules and capabilities
Real Time Control
Physical Machine
Axis 1 i0 Node
Axis 2 i0 Node
Axis 3 i0 Node
…

MATH STRING REPRESENTING A SQUARE

((X >= 0.5) & (X <= 1.5) &

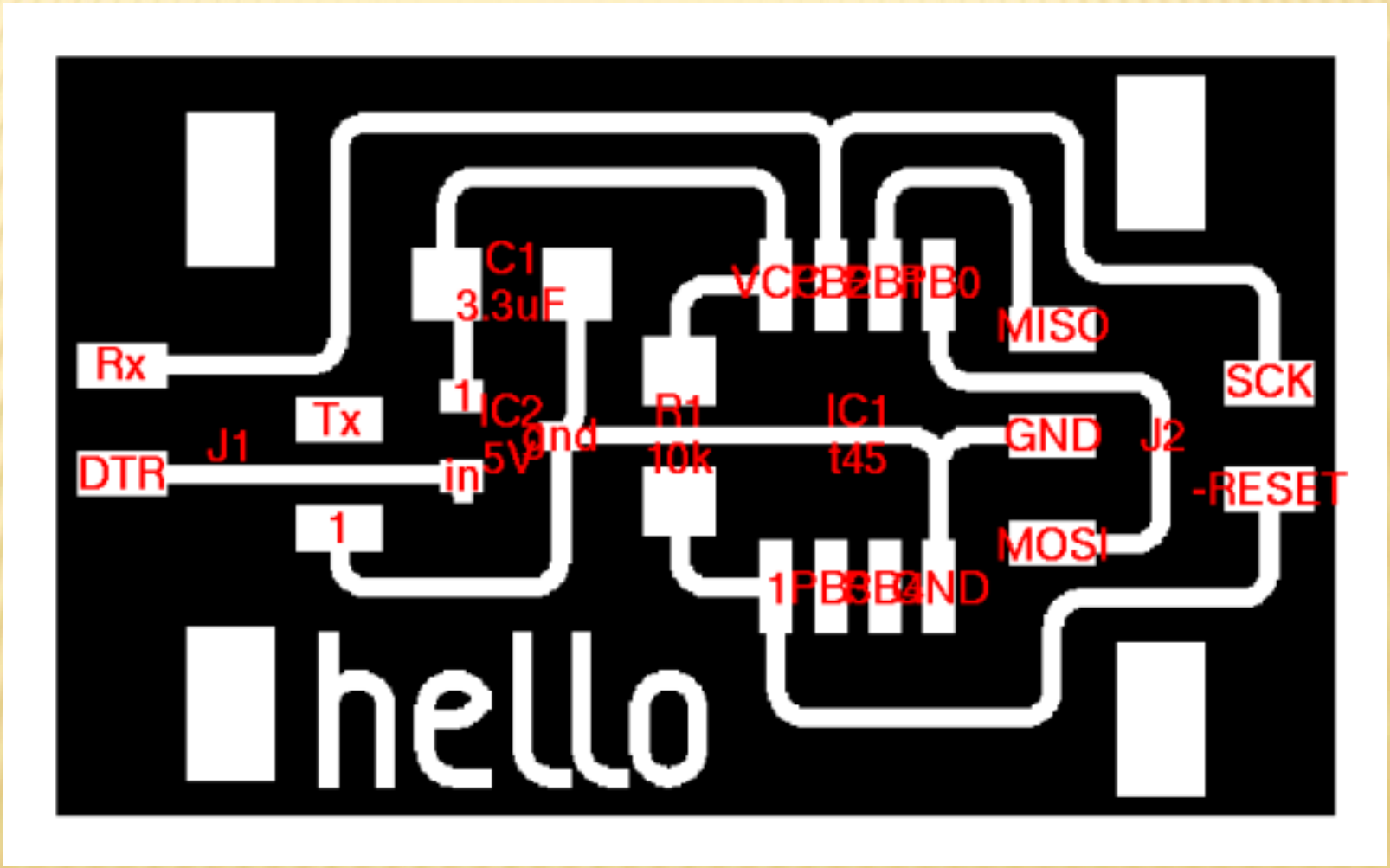
(Y >= 0.5) & (Y <= 1.5))

SLIGHTLY MORE COMPLEX...

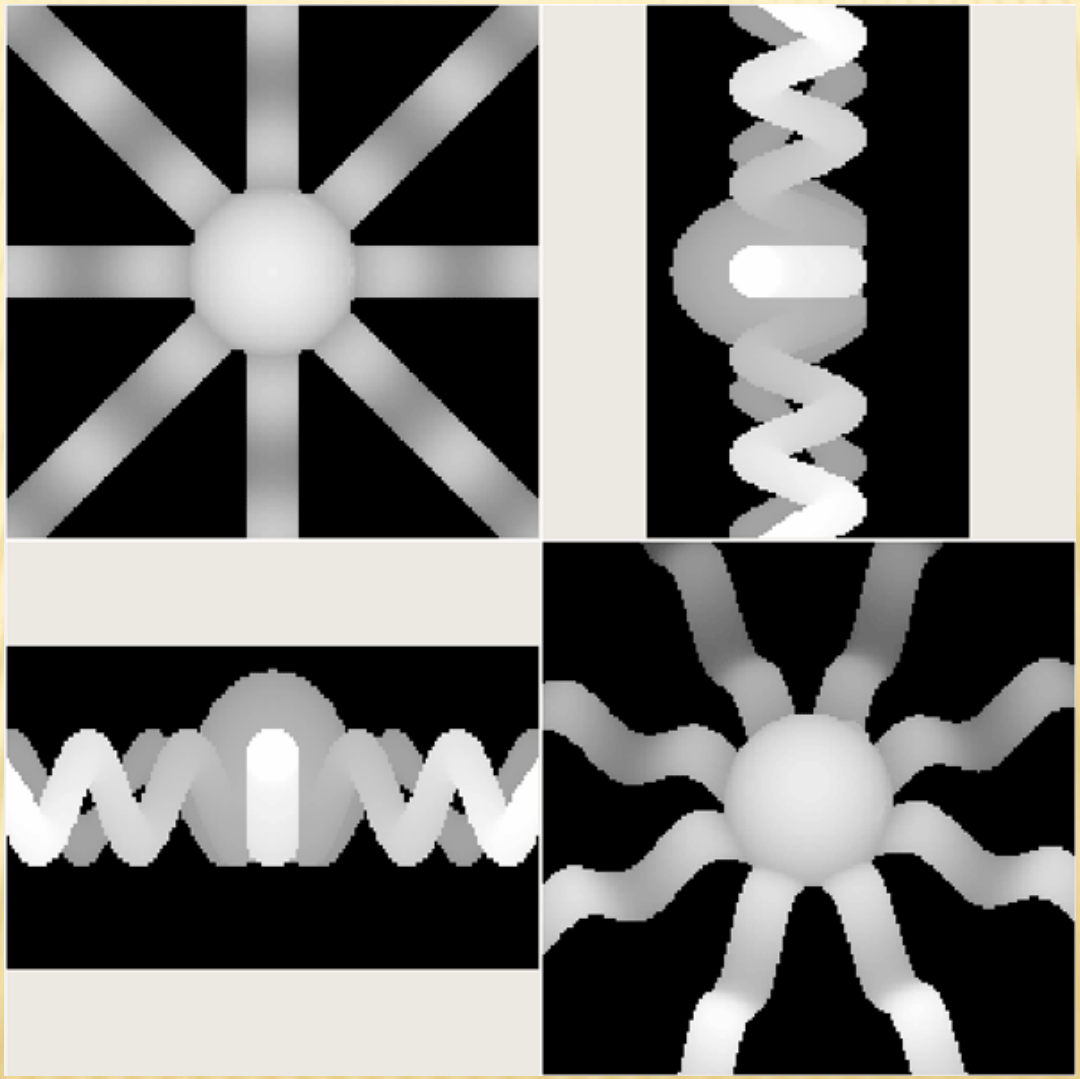
```
((X >= 0) & (X <= 5) & (Y >= 0) & (Y <= 32)) | ((X
   >= 9) & (X <= 14) & (Y >= 10) & (Y <= 32)) | ((X
   >= 18) & (X <= 23) & (Y >= 0) & (Y <= 32)) | ((X
   >= 27) & (X <= 32) & (Y >= 0) & (Y <= 23)) | ((X
   >= 27) & (X <= 32) & (Y >= 27) & (Y <= 32)) |
   ((X >= 36) & (X <= 41) & (Y >= 0) & (Y <= 23)) |
   ((X >= 36) & (X <= 54) & (Y >= 27) & (Y <= 32))
```

...WHICH REPRESENTS:

2D


C1
3.3uF
Rx
Tx
J1
DTR
in
IC2
5V
gnd
R1
10k
IC1
t45
MISO
SCK
GND
J2
-RESET
MOSI
hello

3D



TOO SLOW…

- Bearable for most practical 2D applications
- Hit the limit quickly for 3D objects

REVISIT THE MATH STRING FOR A SQUARE

((X >= 0.5) & (X <= 1.5) &

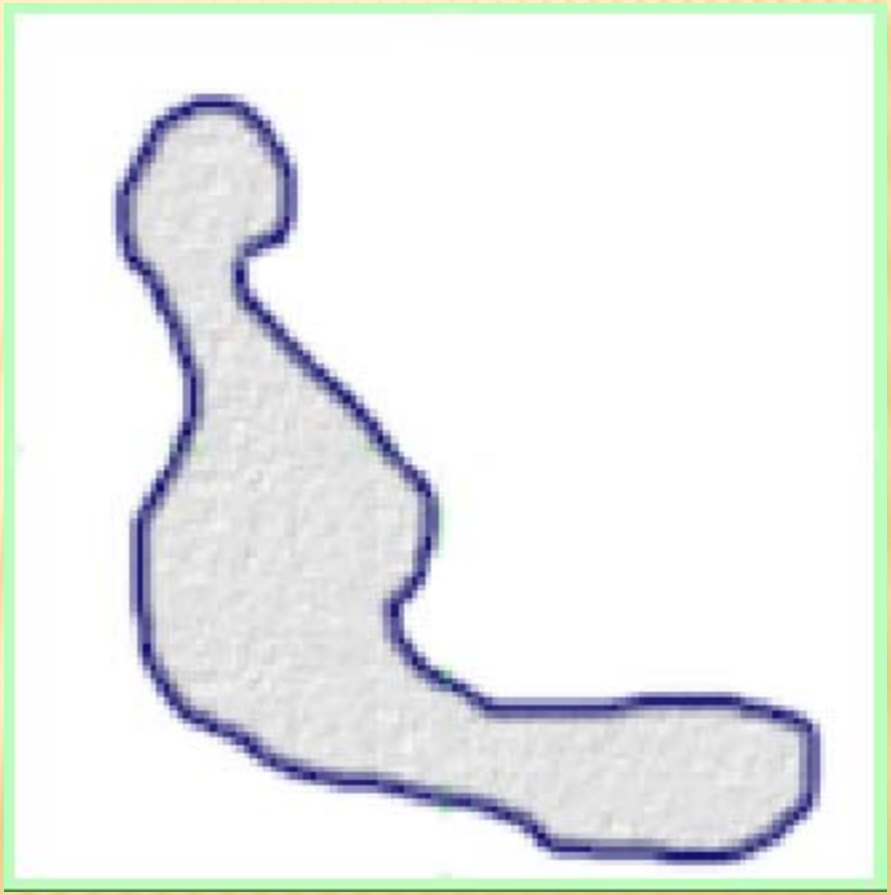
(Y >= 0.5) & (Y <= 1.5))

IMPROVEMENTS

- Don’t perform the evaluation everywhere
- Don’t evaluate the entire expression

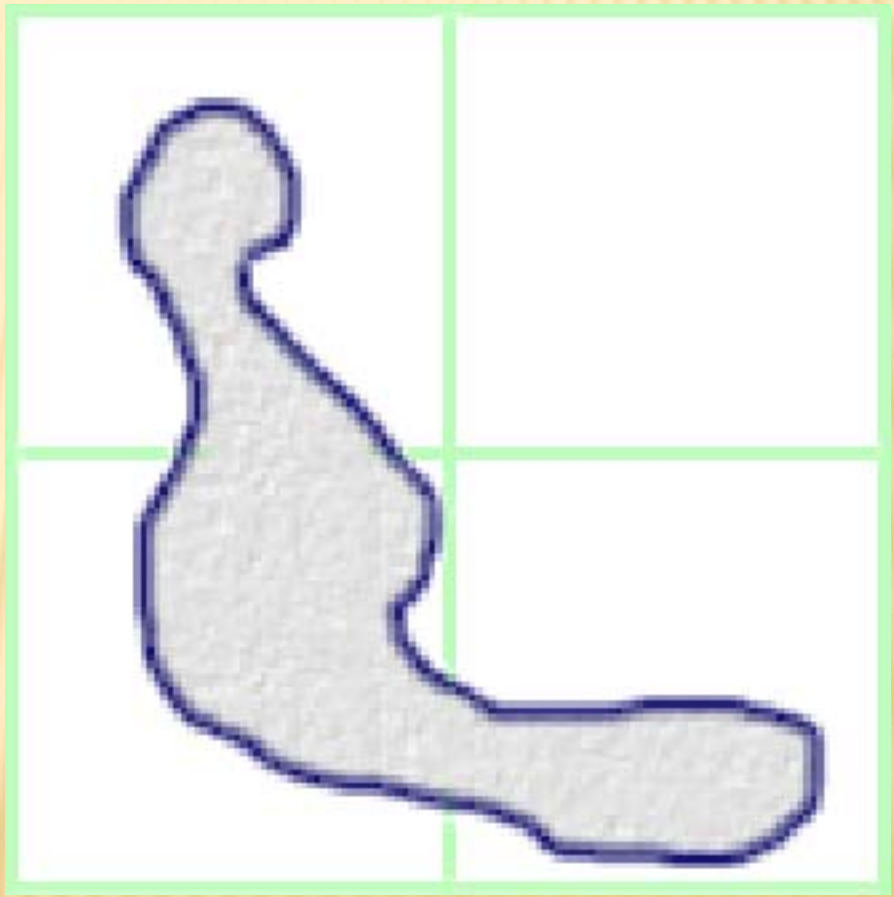
OCTREE / QUADTREE

An object:



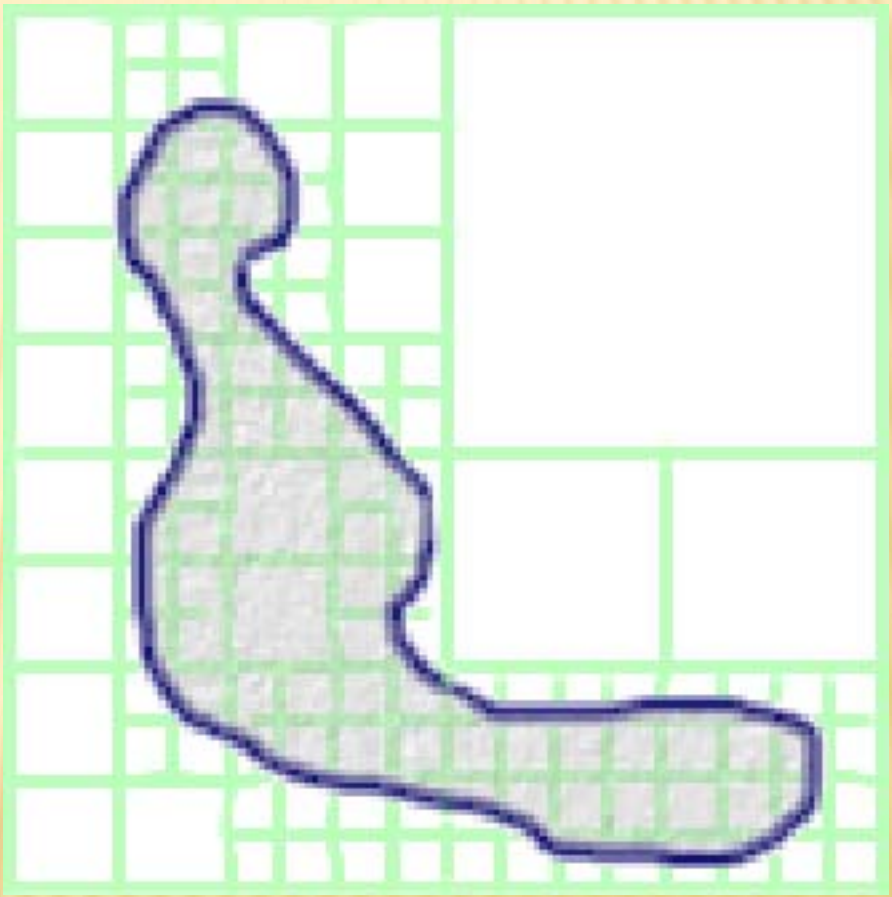
OCTREE / QUADTREE

Start chopping:



OCTREE / QUADTREE

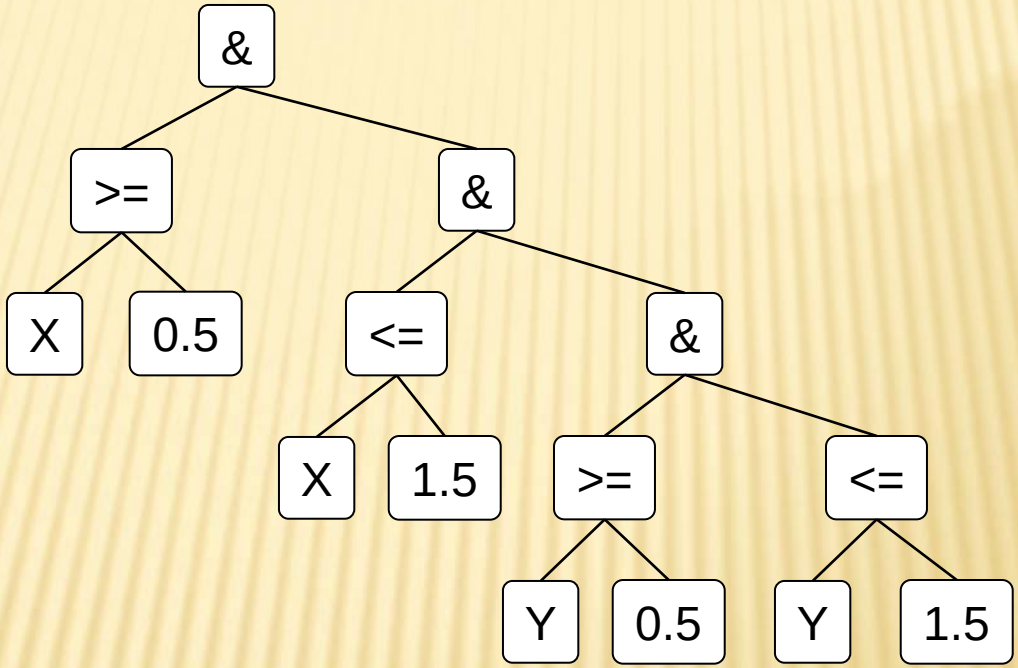
Recursively:



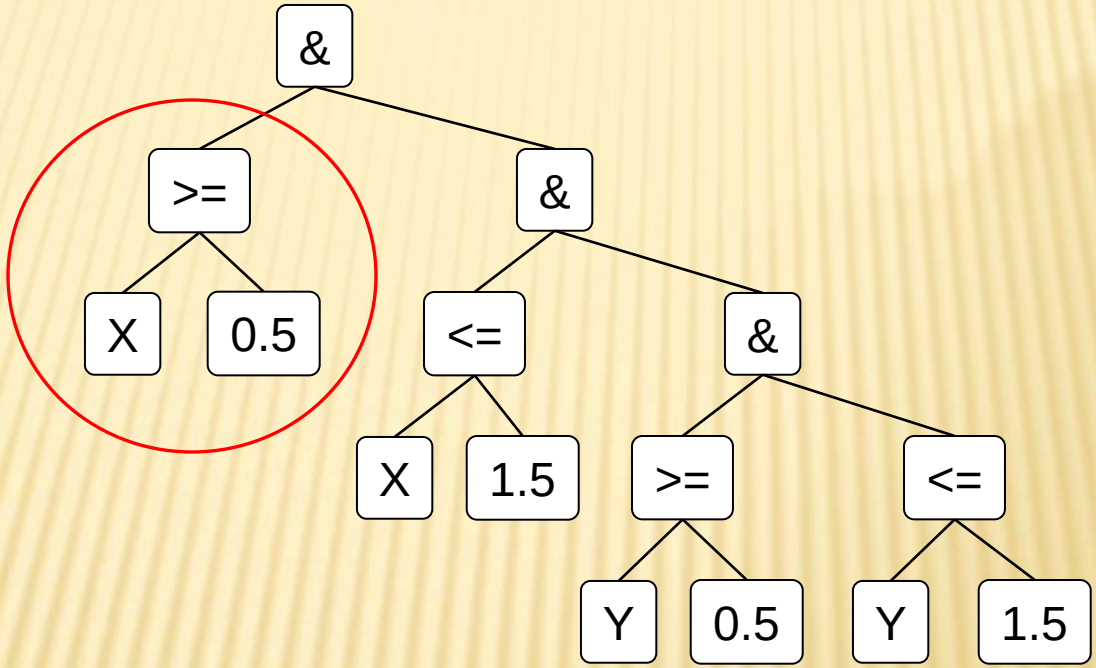
But without evaluating the math string everywhere, we don't yet know where the boundaries are…

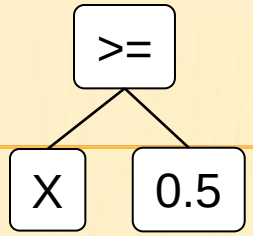
…remind me to answer that question 3 slides from now.

EXPRESSION TREE



EXPRESSION TREE





- Subregions that only contain space in the X>=0.5 portion of the plane do not need to be resolved any further.
- Subregions that span the X=0.5 line may (but are not guaranteed to) contain boundaries between solid material and empty space, so they do need to be resolved further.
- Subregions that only contain space in the X>0.5 portion of the plane may contain boundaries and need to be resolved further, but for these areas, the math string can be pruned to a shorter expression, eliminating the first condition.

RESULT

- Our quadtree/octree shows us large swaths of area that can be ignored
- We do not have to evaluate the entire math string anywhere. Significant chunks are pruned away

WHAT IF OUR SQUARE GETS A LITTLE CURVY?

```
( ( X >= sin(Y/(2*pi)) ) & (X <= 1.5) &
      (Y >= 0.5) & (Y <= 1.5))
```

EXPRESSION INCLUDES A RANGE

((X >= some value in the range [-1,1]) & (X <= 1.5) & (Y >= 0.5) & (Y <= 1.5))

IMPLICATIONS

- Use interval arithmetic to deal with compound operations on ranges
- Slightly wider area in which the octree must be finely resolved

GENERATING TOOLPATHS

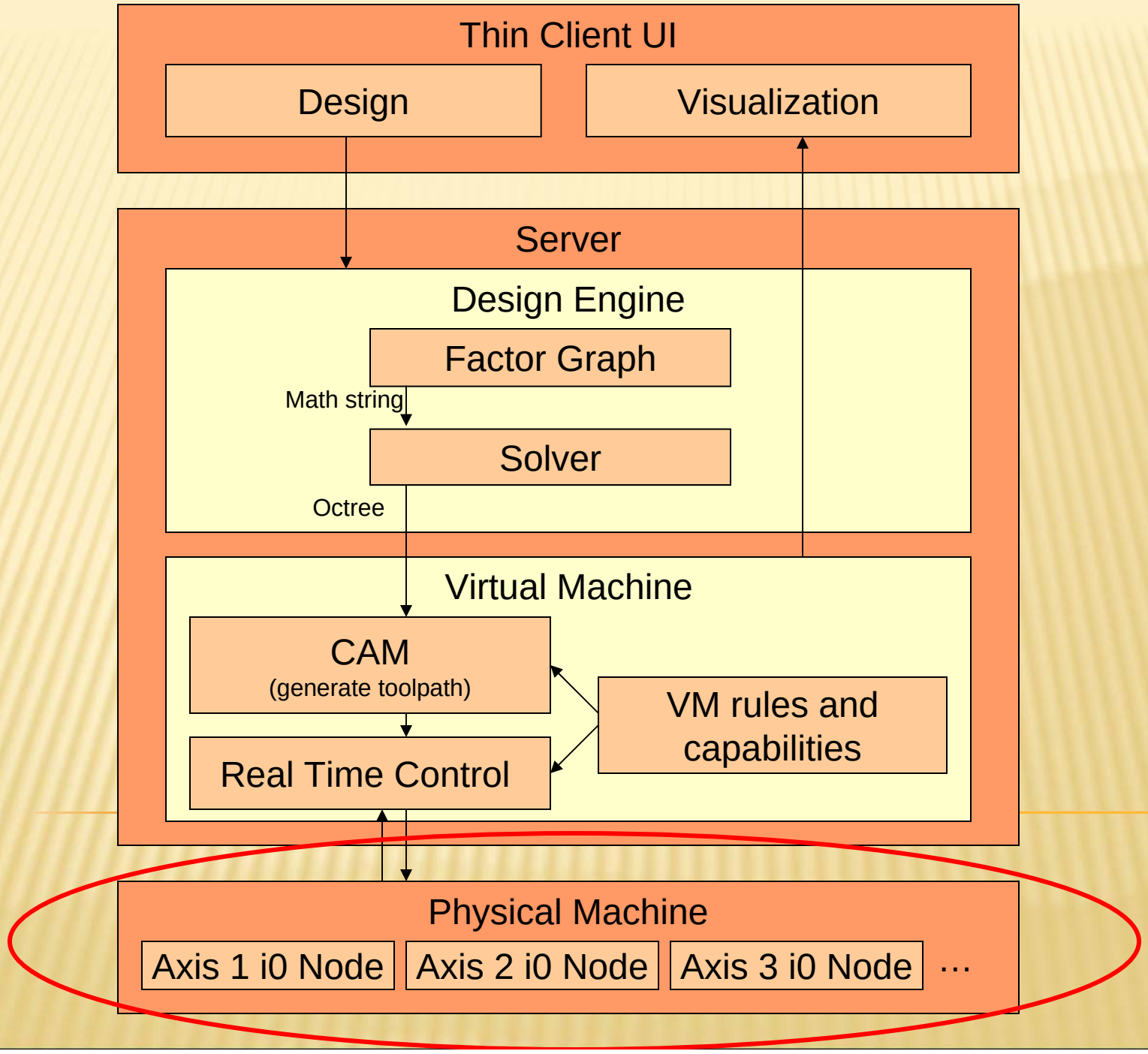
- Was easy when we had a fully resolved grid of values indicating True/False for the presence of material
- Do something analogous on the octree

GENERATING TOOLPATHS

- Evaluate (pruned) math string in octree leaf node region
- Compare with neighbors, identify boundaries
- Fully resolve to specified resolution along the boundary and where contours are requested
- Ignore everything else

TOOLPATH FORMAT (INSTRUCTIONS FOR THE VIRTUAL MACHINE)

```
traverse_speed = 8
cutting_speed = 4.0
plunge_speed = 4.0
z_down = 0.0
z_up = 0.1
move(z=z_up, rate=plunge_speed)
move(0.0, 0.0, z_up, traverse_speed)
move(z=z_down, rate=plunge_speed)
move(0.0462962963, 0.0, z_down, cutting_speed)
move(0.0462962963, 0.29629629629, z_down,
    cutting_speed)
...
```


Thin Client UI
Design
Visualization
Server
Design Engine
Factor Graph
Math string
Solver
Octree
Virtual Machine
CAM
(generate toolpath)
Real Time Control
VM rules and capabilities
Physical Machine
Axis 1 i0 Node
Axis 2 i0 Node
Axis 3 i0 Node
…

…and then the exciting part happens.

Things start getting built.