Programming Environments for

Generative Design

Guilherme Ferreira

Instituto Superior Técnico

1. Problem
2. Survey
3. Solution
4. Conclusion

1

Problem

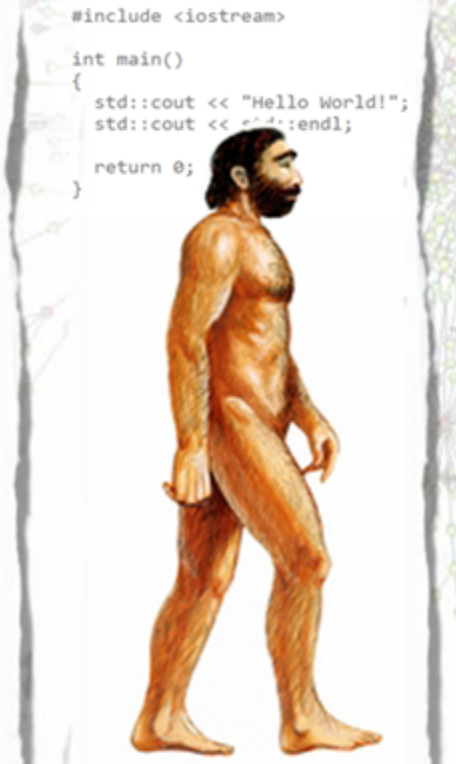
The Evolution of Computer Programming Languages

Binary

Assembly

Fortran

C

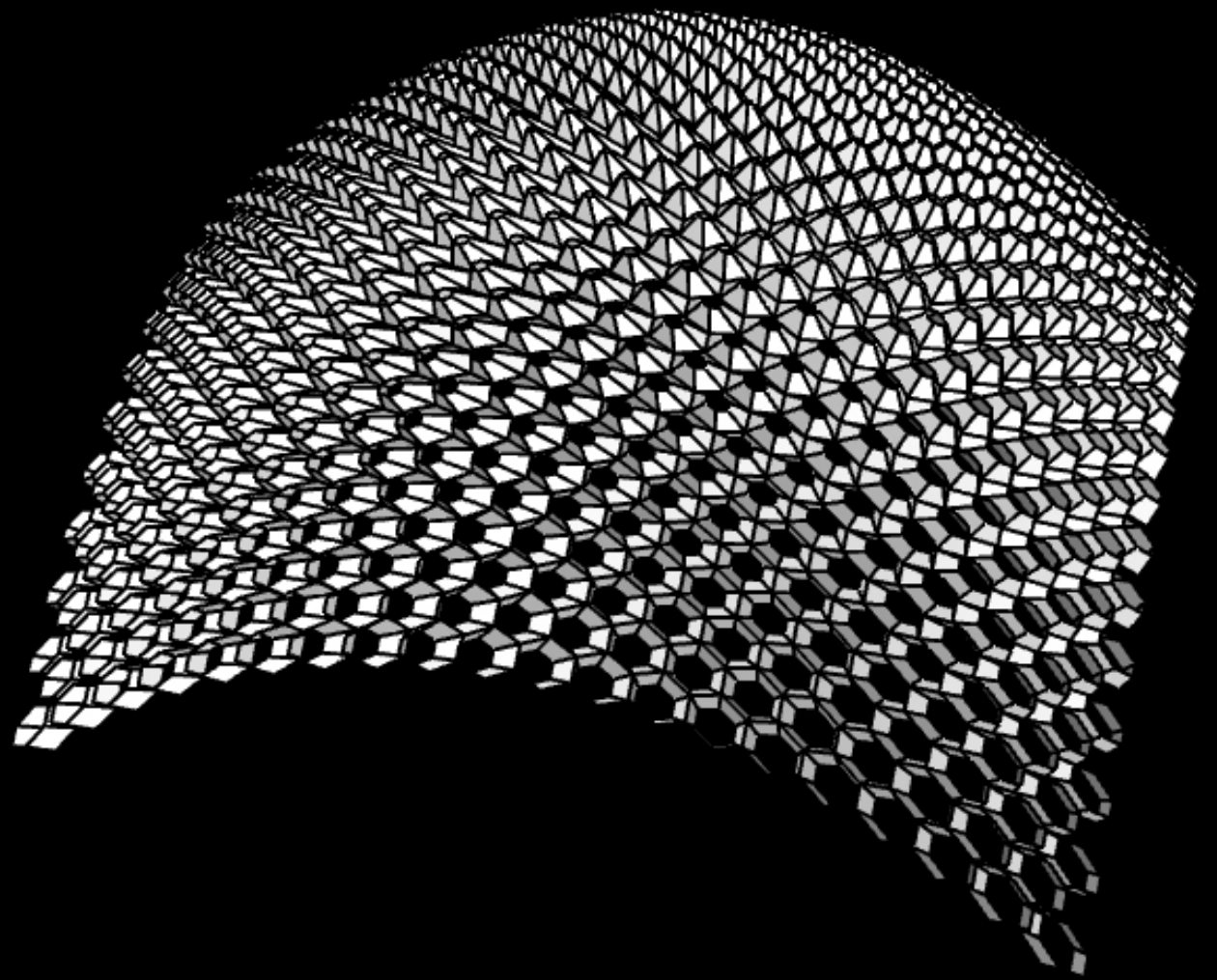


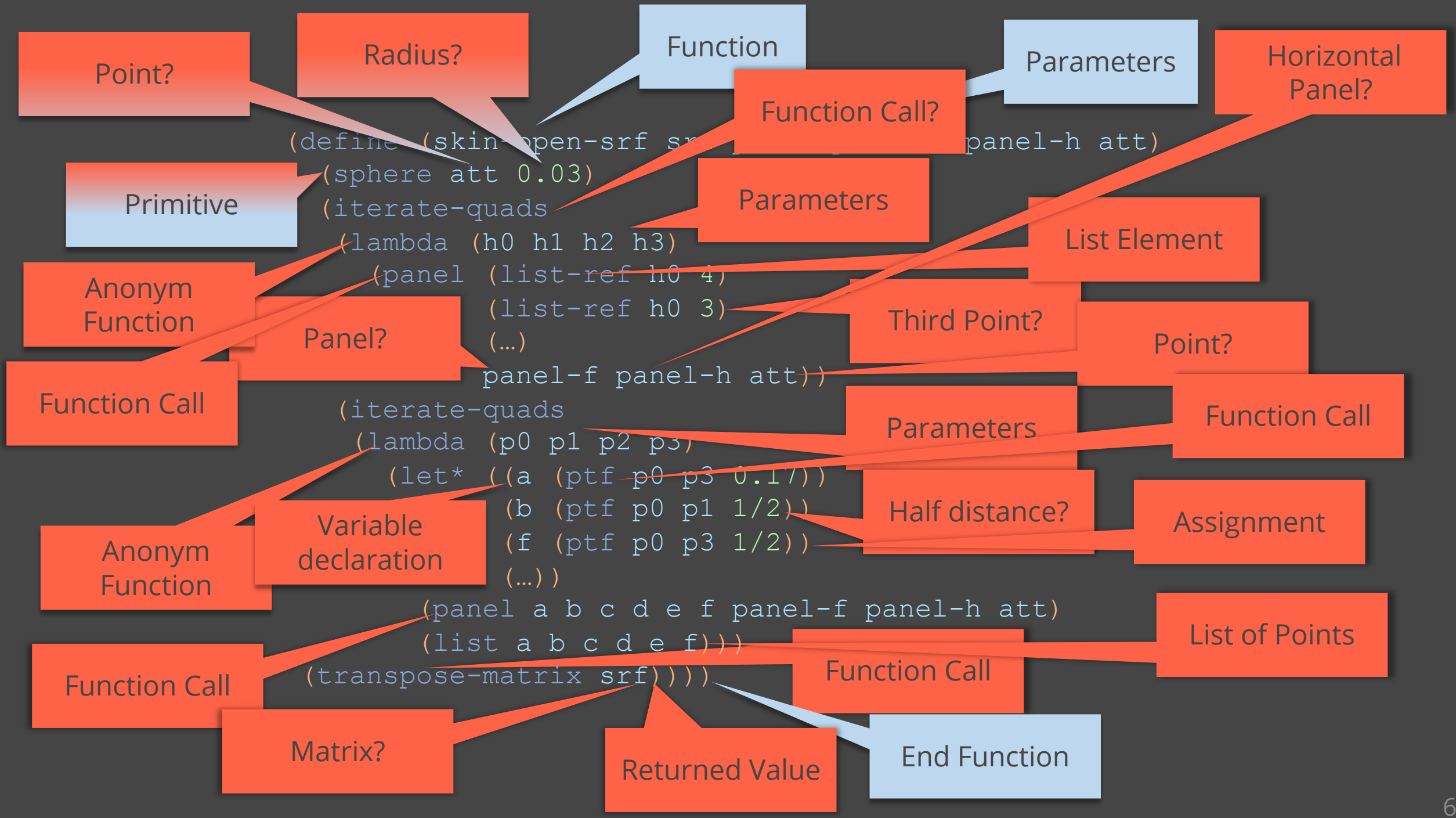
C++

Java

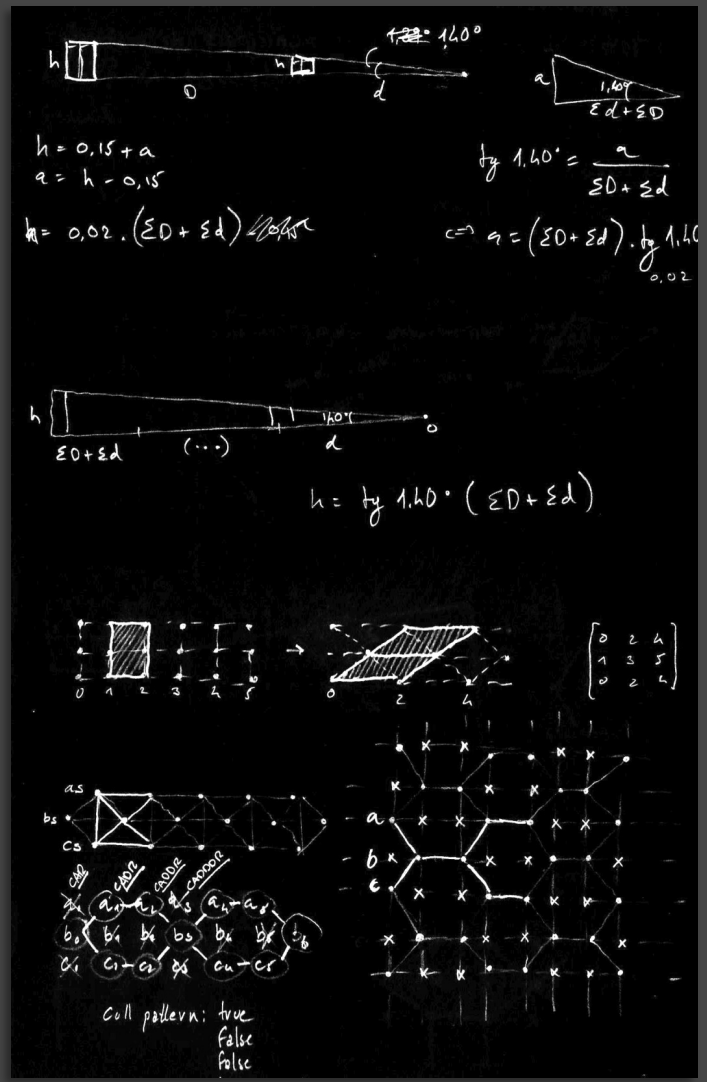
```
(define (skin-open-srf srf panel panel-f panel-h att)
  (sphere att 0.03)
  (iterate-quads
   (lambda (h0 h1 h2 h3)
     (panel (list-ref h0 4)
            (list-ref h0 3)
            (list-ref h1 4)
            (list-ref h2 1)
            (list-ref h2 0)
            (list-ref h3 1)
           panel-f panel-h att))
   (iterate-quads
    (lambda (p0 p1 p2 p3)
      (let* ((a (ptf p0 p3 0.17))
             (b (ptf p0 p1 1/2))
             (f (ptf p0 p3 1/2))
             (j (ptf p3 p2 1/2))
             (e (ptf b j 2/3))
             (k p1)
             (l p2)
             (c (ptf k l 0.17))
             (d (ptf k l 1/2))
             (w (ptf k l 2/3))
             (g (ptf (ptf p0 p3 2/3) (ptf p1 p2 2/3) 3/2))
             (h (ptf p3 p2 3/2))
             (i (ptf k l 1.17)))
        (panel a b c d e f panel-f panel-h att)
        (list a b c d e f)))
    (transpose-matrix srf))))

(define (hex-panel-1 p0 p1 p2 p3 p4 p5 f h att)
  ...)

(define (hex-panel-2 p0 p1 p2 p3 p4 p5 f h att)
  ...)
```

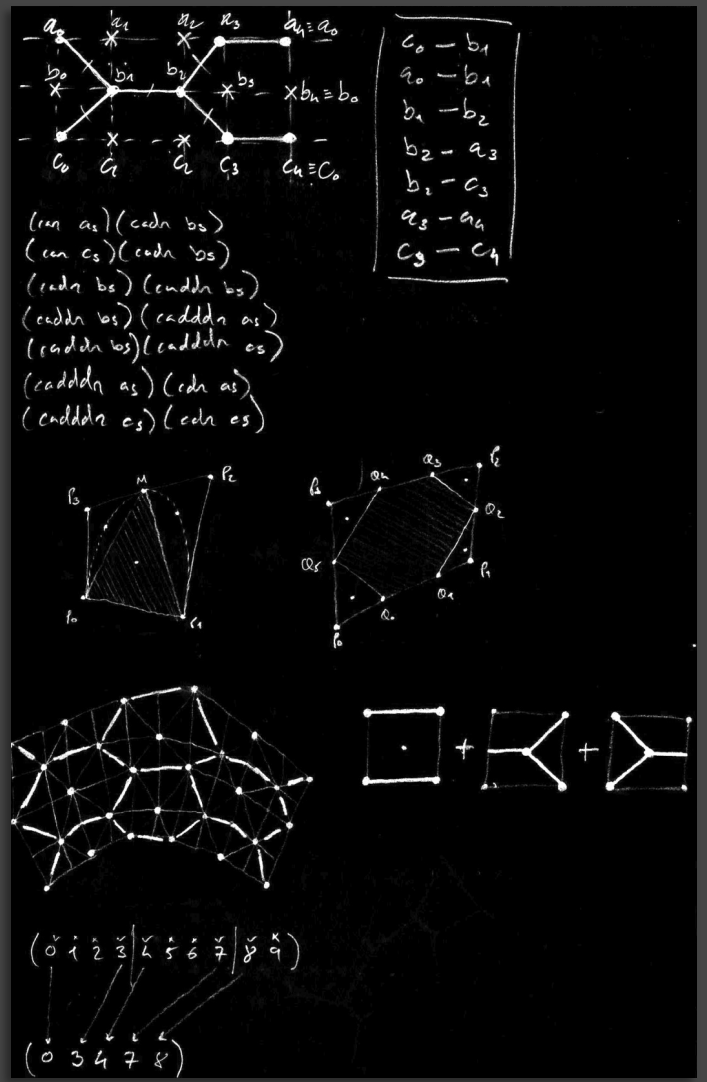
Generative Design Environments

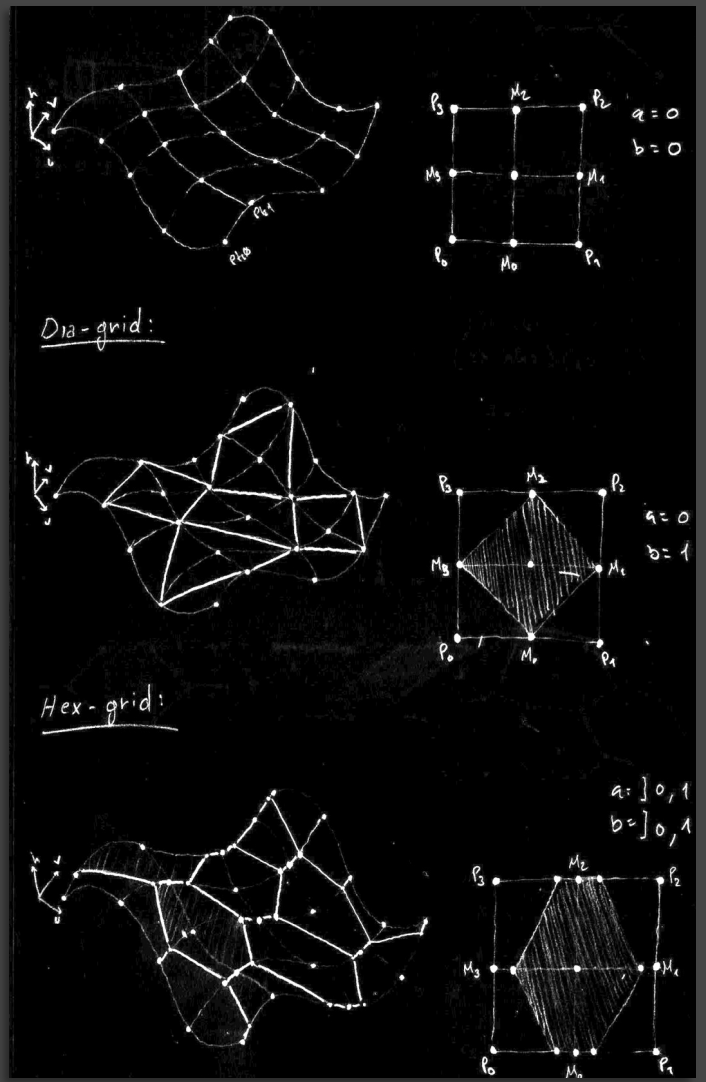



Point?
Radius?
Function
Parameters
Horizontal Panel?
Function Call?
(define (skin-open-srf sr panel-h att)
(sphere att 0.03)
(iterate-quads
Primitive
Parameters
(lambda (h0 h1 h2 h3)
List Element
Anonym Function
(panel (list-ref h0 4)
(list-ref h0 3)
Third Point?
Point?
Panel?
(…)
Function Call
panel-f panel-h att))
(iterate-quads
Parameters
Function Call
(lambda (p0 p1 p2 p3)
(let* ((a (ptf p0 p3 0.17))
Anonym Function
Variable declaration
(b (ptf p0 p1 1/2))
Half distance?
Assignment
(f (ptf p0 p3 1/2))
(…))
(panel a b c d e f panel-f panel-h att)
List of Points
(list a b c d e f)))
Function Call
Function Call
(transpose-matrix srf))))
Matrix?
Returned Value
End Function

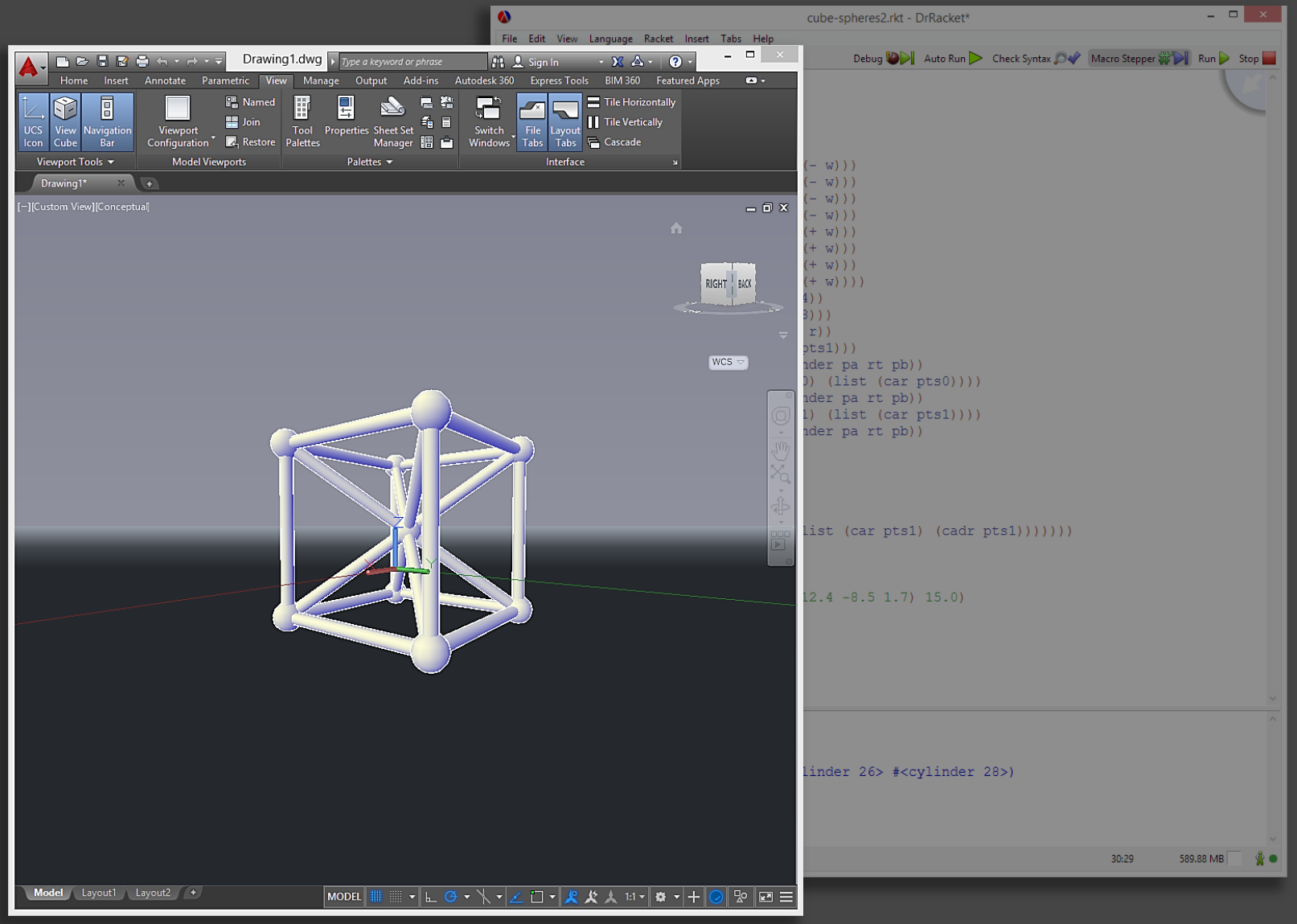
People understand what they can see



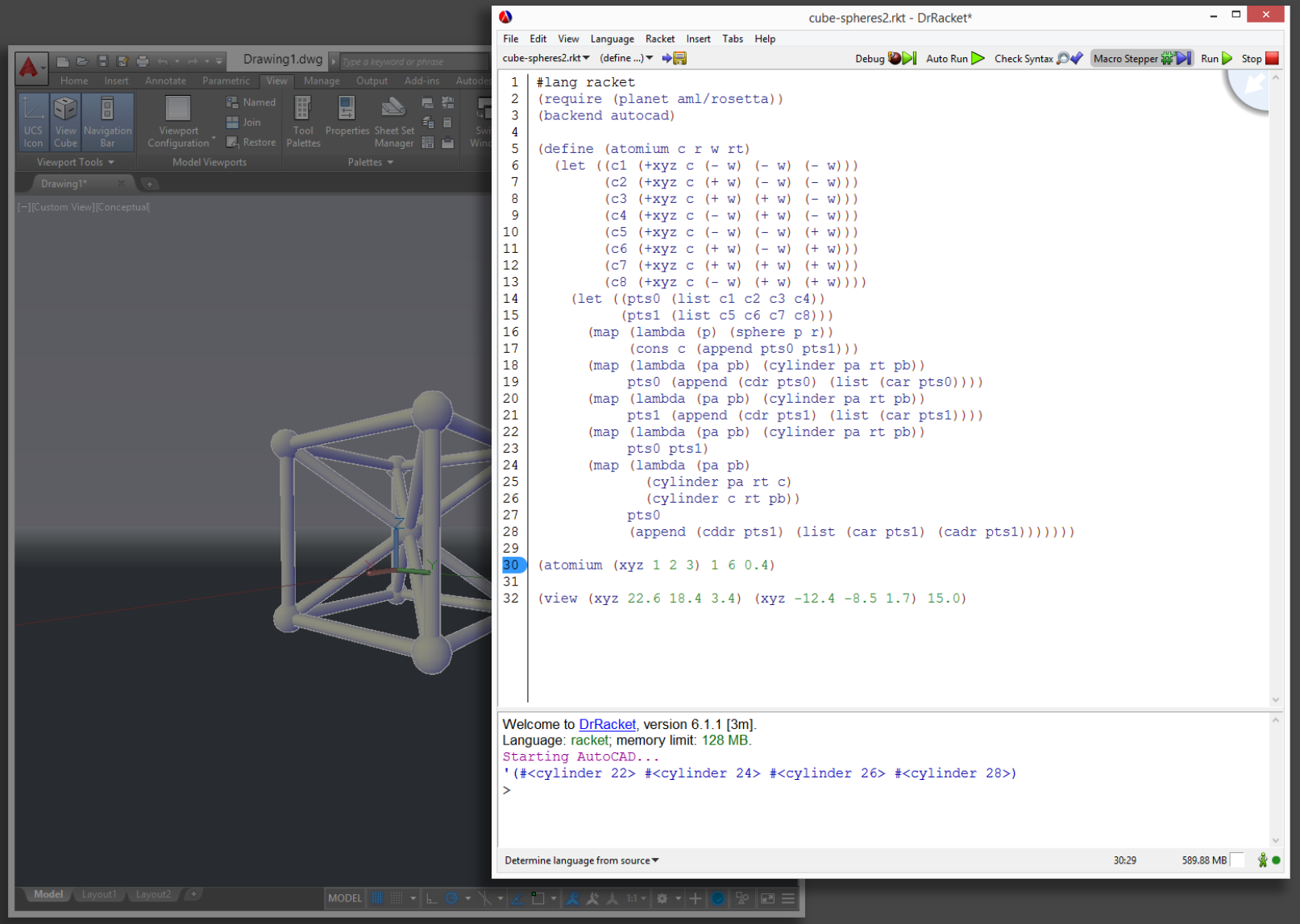




Creators need a medium connection



Creators need a medium connection



2

Survey

LightTable (Chris Granger et al)

cube.js*

```
//

function animate() {

    requestAnimationFrame( animate );

    render();
    stats.update();

}

function render() {

    var time = Date.now() * 0.

    mesh.rotation.x = time *
    mesh.rotation.y = time * 1.5;

    renderer.render( scene, camera );

}
```

12
16
160000
21
001
21845
light1
Uint16Array

three.js webgl - buffergeometry

60 FPS (7-62)

three.js webgl - buffergeometry

Live

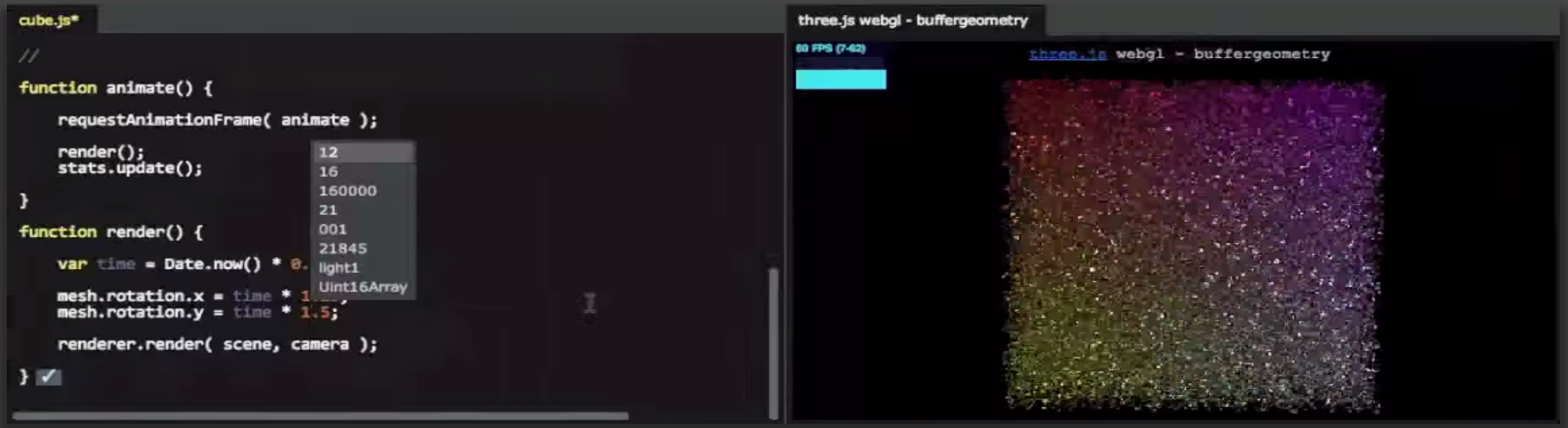
```
(apply + [1 2 3 4])

(defn hello-light [msg]
  (str "Message: " msg))

(hello-light "hello")
```

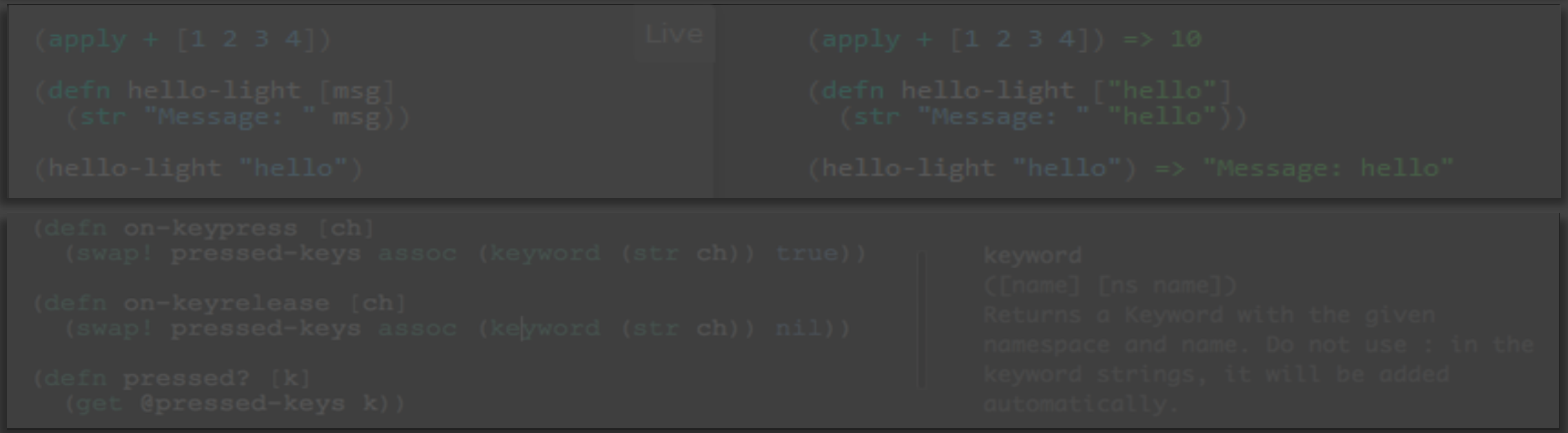
```
(apply + [1 2 3 4]) => 10

(defn hello-light ["hello"]
  (str "Message: " "hello"))

(hello-light "hello") => "Message: hello"
```

```
(defn on-keypress [ch]
  (swap! pressed-keys assoc (keyword (str ch)) true))

(defn on-keyrelease [ch]
  (swap! pressed-keys assoc (keyword (str ch)) nil))

(defn pressed? [k]
  (get @pressed-keys k))
```

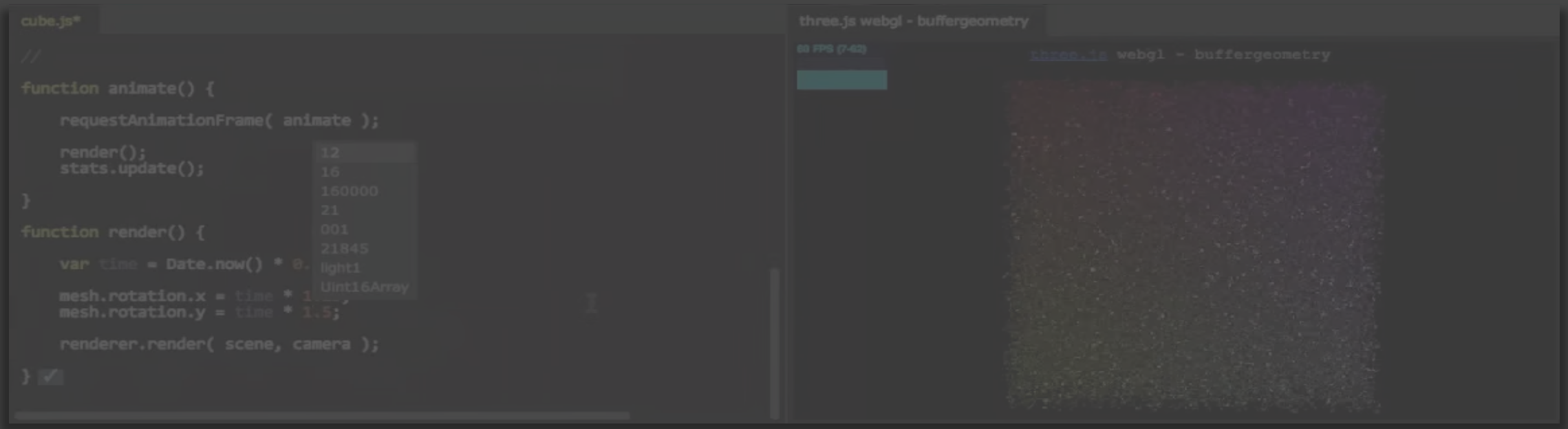
```
keyword
([name] [ns name])
Returns a Keyword with the given
namespace and name. Do not use : in the
keyword strings, it will be added
automatically.
```

LightTable (Chris Granger et al)





LightTable (Chris Granger et al)




```
(apply + [1 2 3 4])

(defn hello-light [msg]
  (str "Message: " msg))

(hello-light "hello")
```

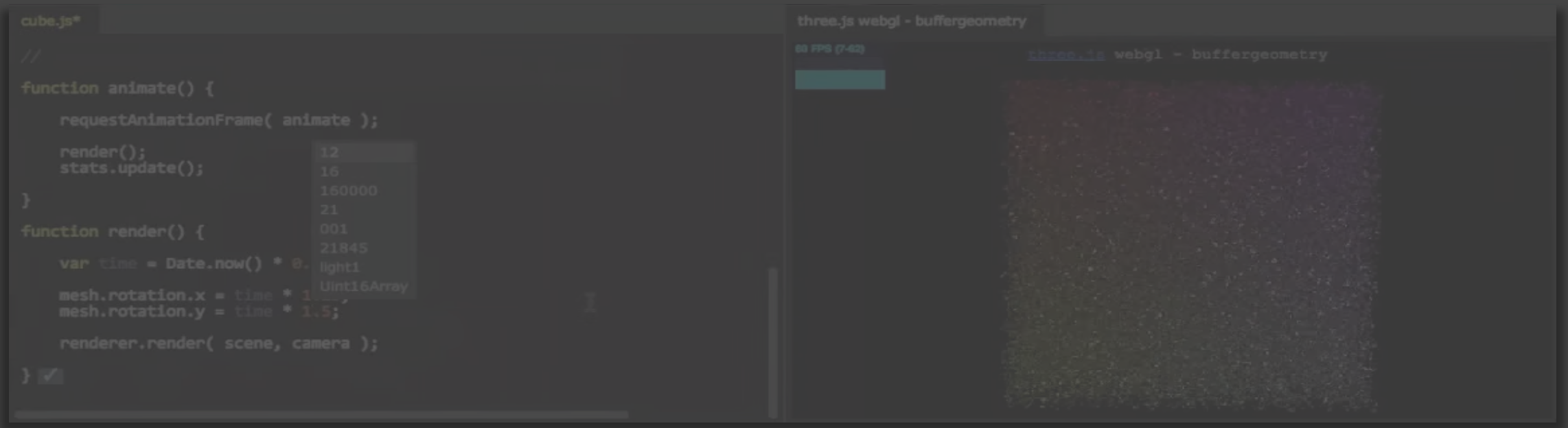
Live

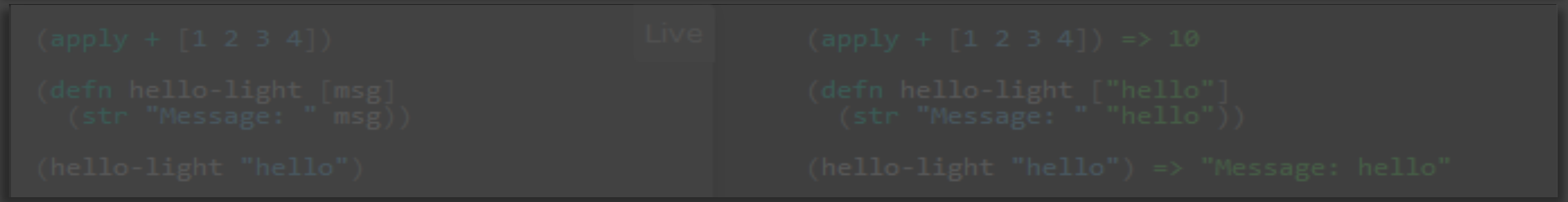
```
(apply + [1 2 3 4]) => 10

(defn hello-light ["hello"]
  (str "Message: " "hello"))

(hello-light "hello") => "Message: hello"
```

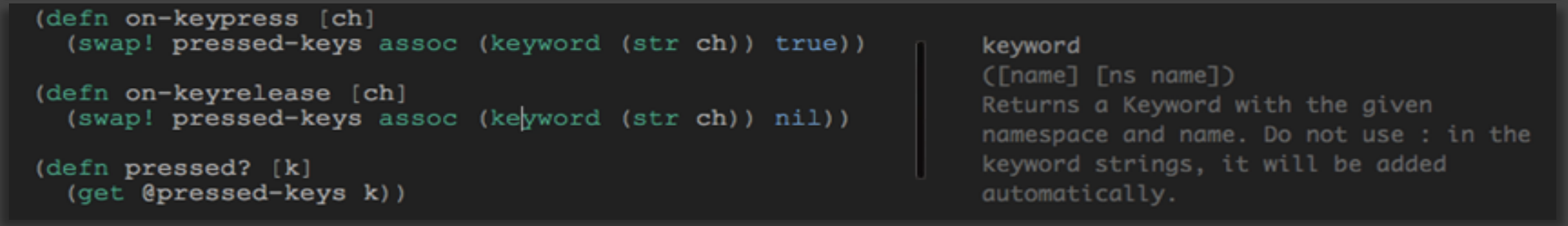
```
(defn on-keypress [ch]
  (swap! pressed-keys assoc (keyword (str ch)) true))

(defn on-keyrelease [ch]
  (swap! pressed-keys assoc (keyword (str ch)) nil))

(defn pressed? [k]
  (get @pressed-keys k))
```

```
keyword
([name] [ns name])
Returns a Keyword with the given
namespace and name. Do not use : in the
keyword strings, it will be added
automatically.
```

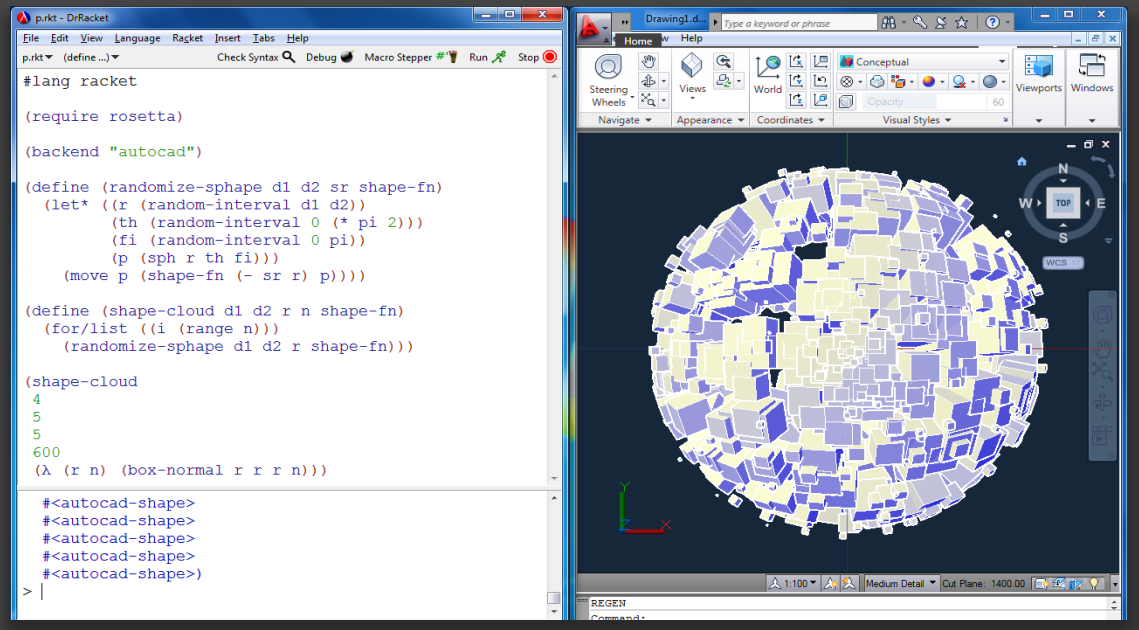
LightTable (Chris Granger et al)



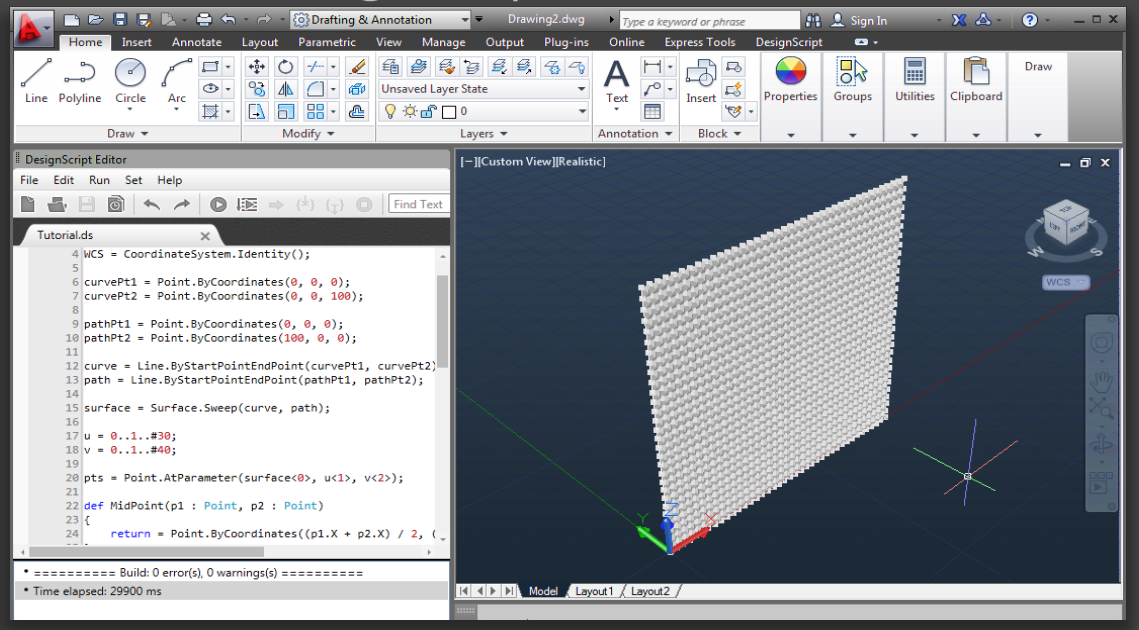




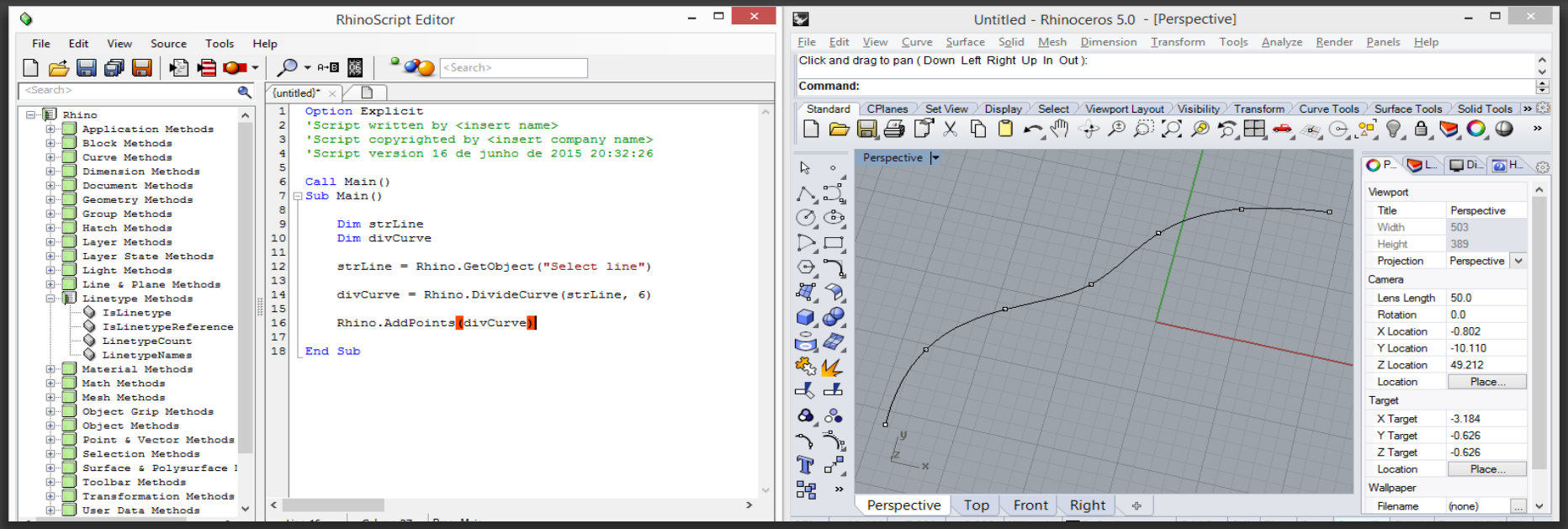
Rosetta (António Leitão et al)



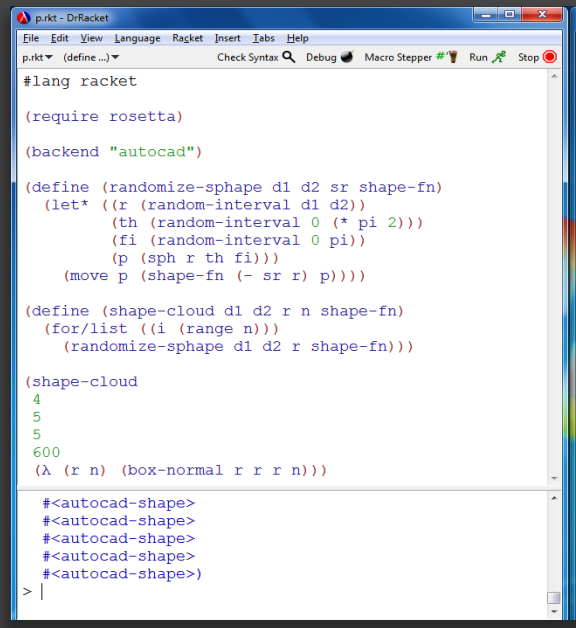
DesignScript (Robert Aish et al)

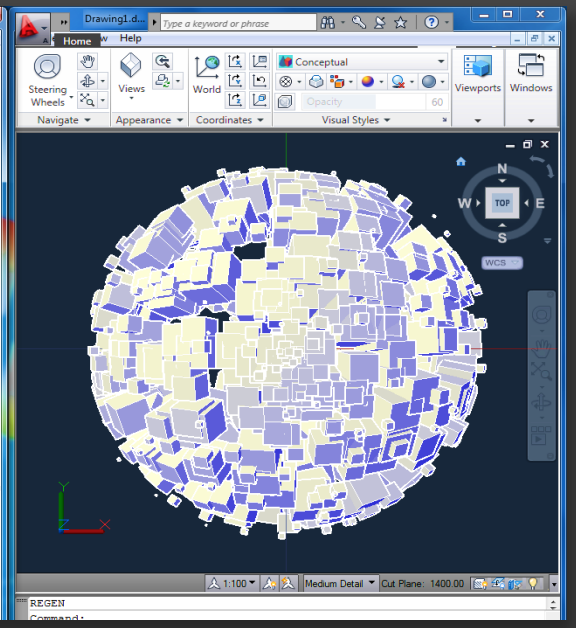


Monkey (David Rutten et al)

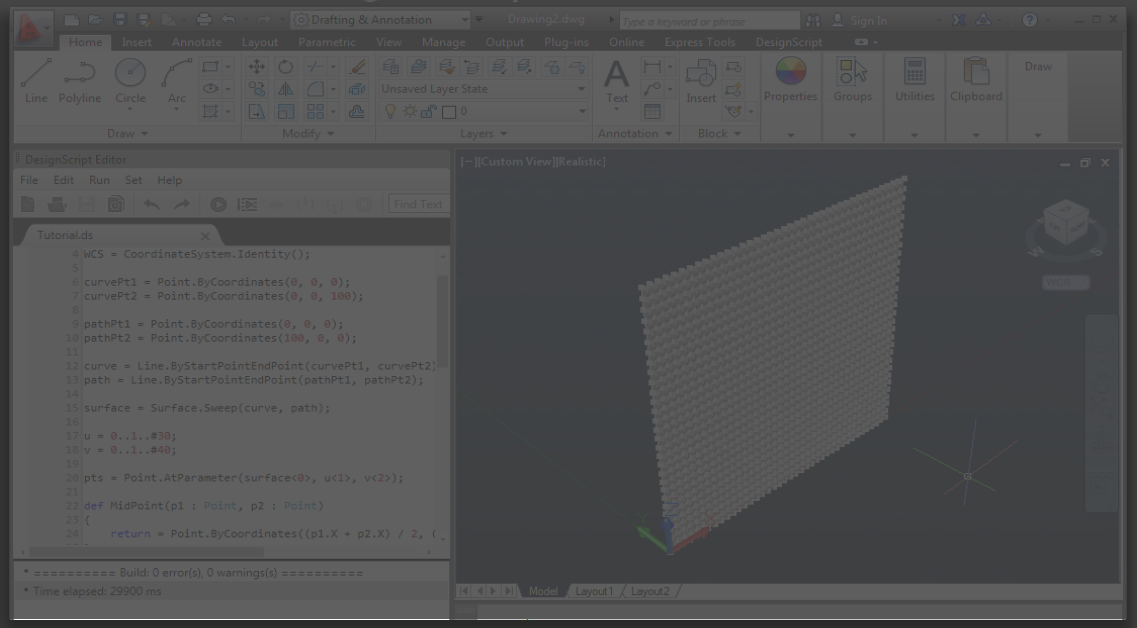


Rosetta (António Leitão et al)

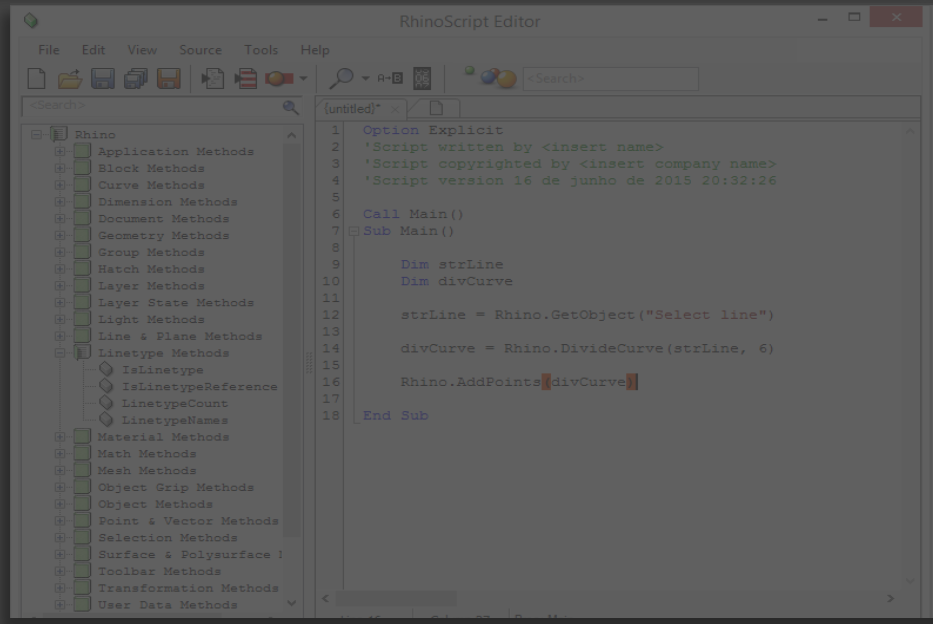


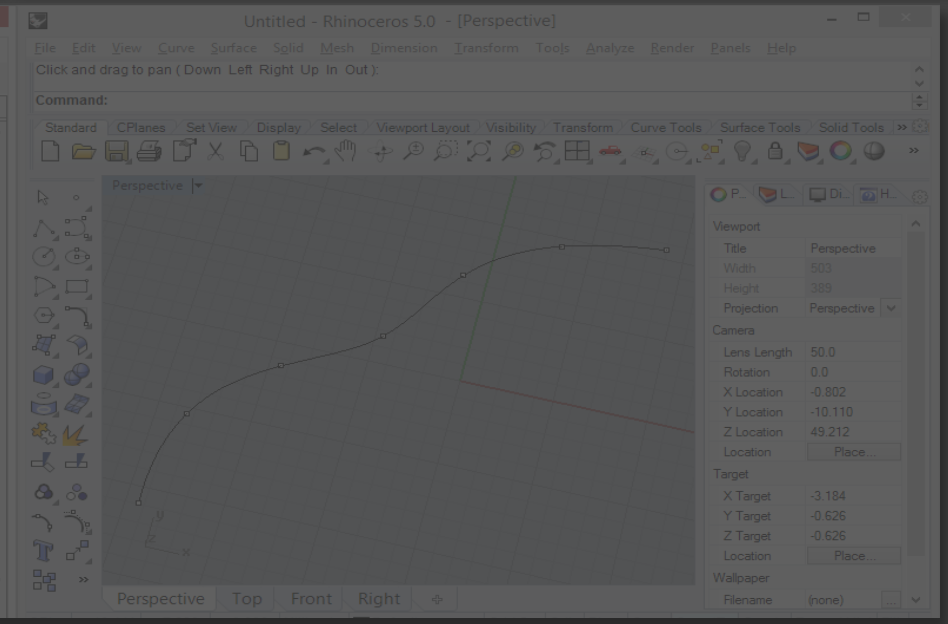


DesignScript (Robert Aish et al)

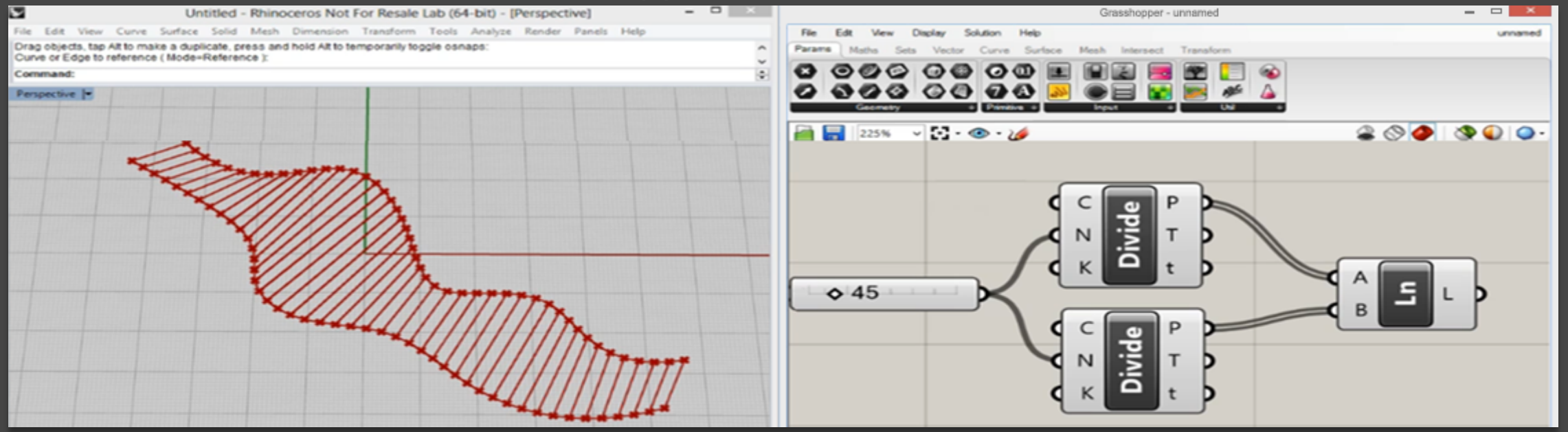


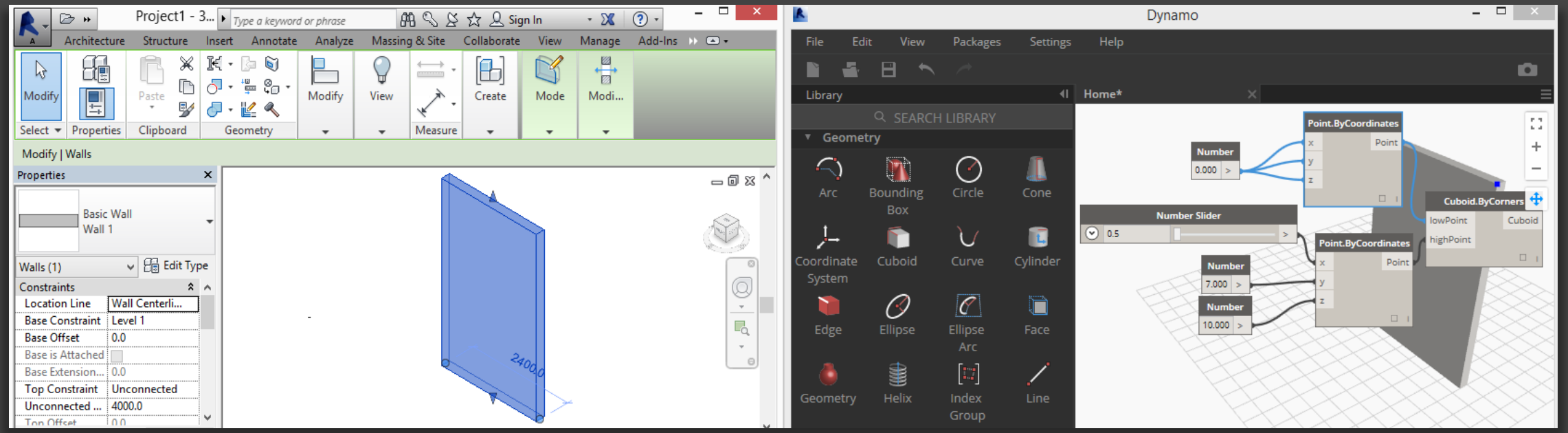
Monkey (David Rutten et al)





Grasshopper (David Rutten et al)





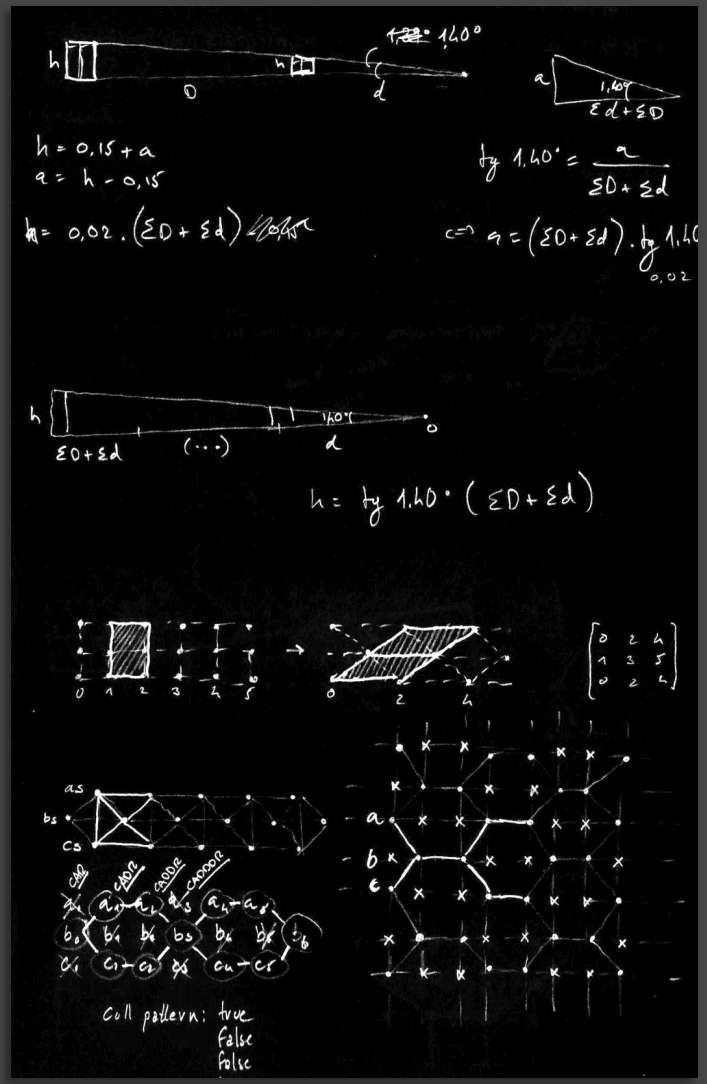
Dynamo (AutoDesk)

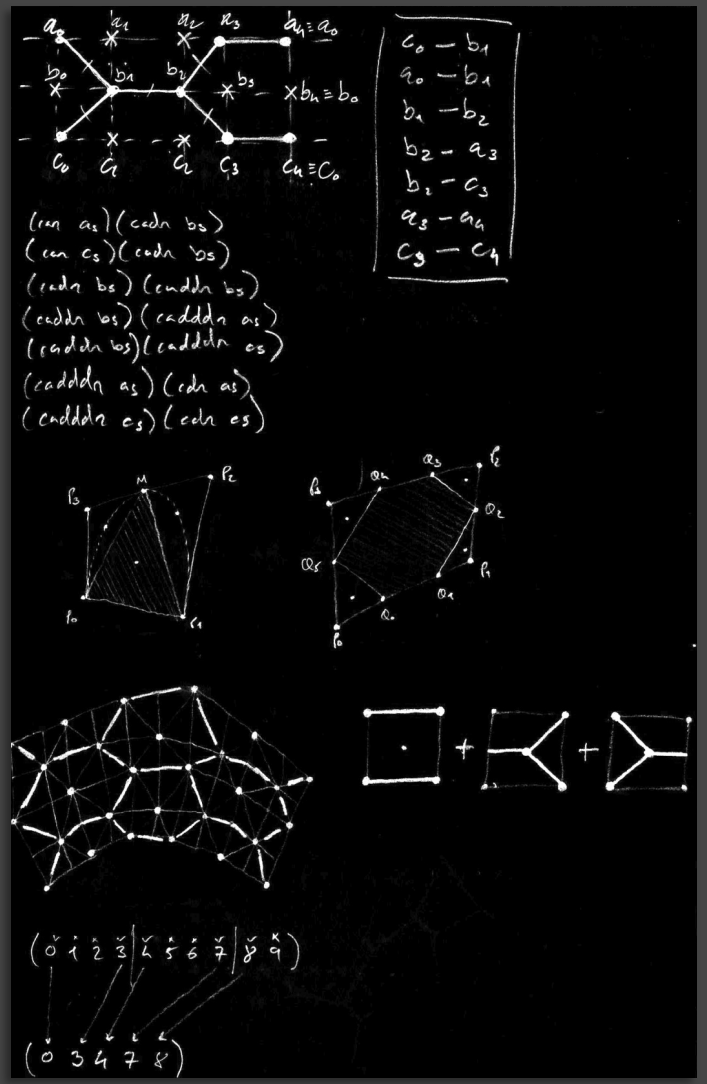
How can we understand this program?

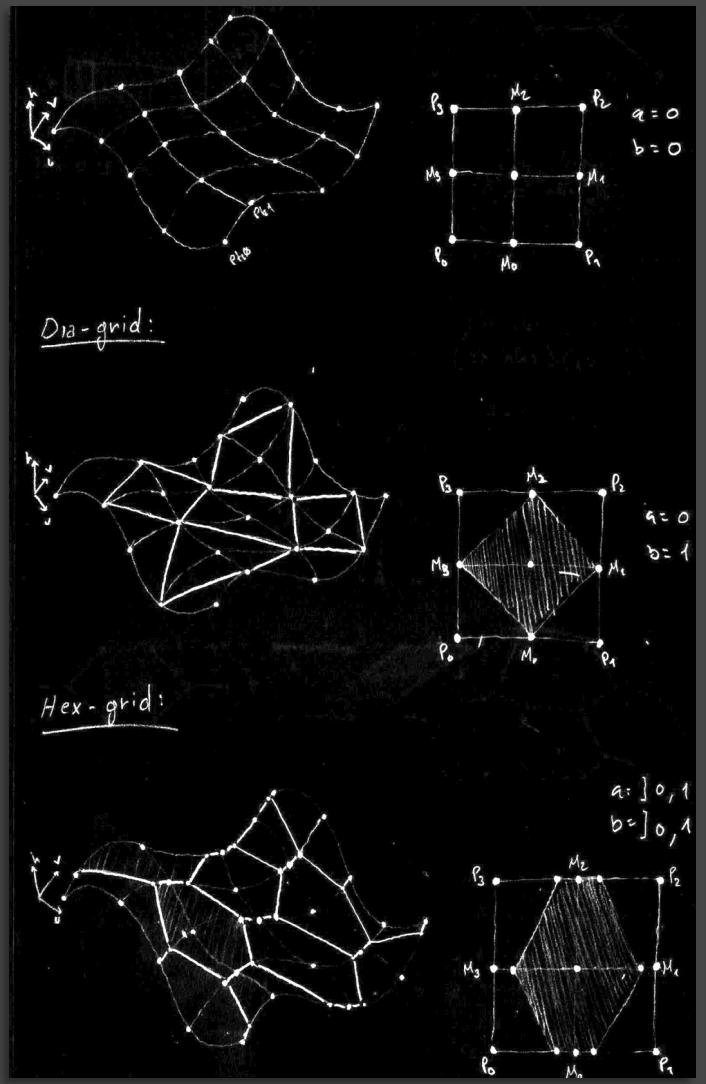
3

Solution

People understand what they can see

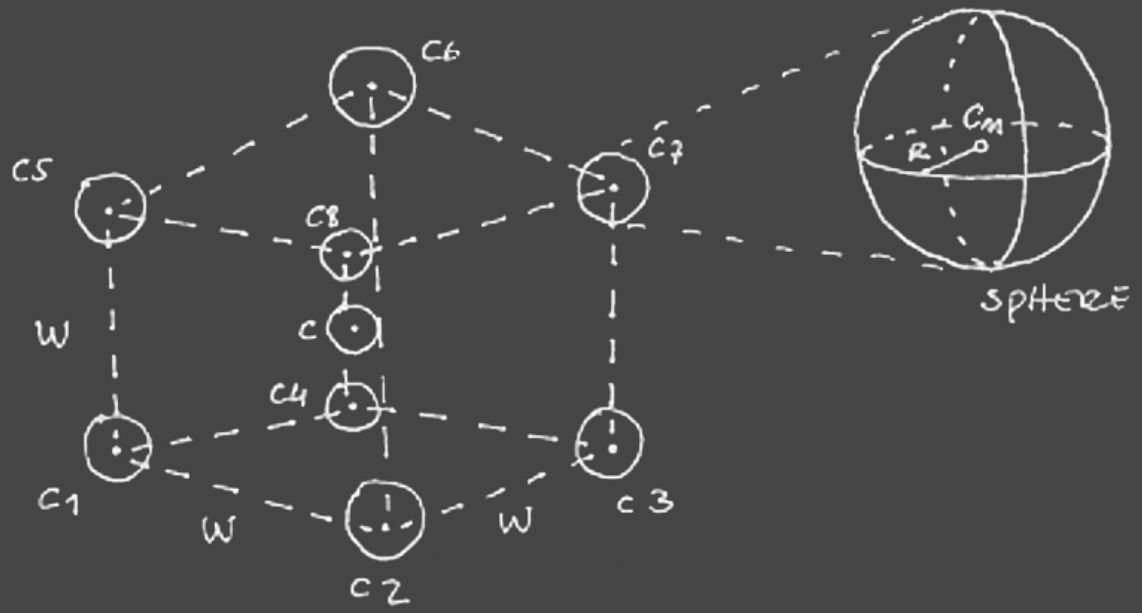






Program

```
def cubeSpheres(c, w, r):
```

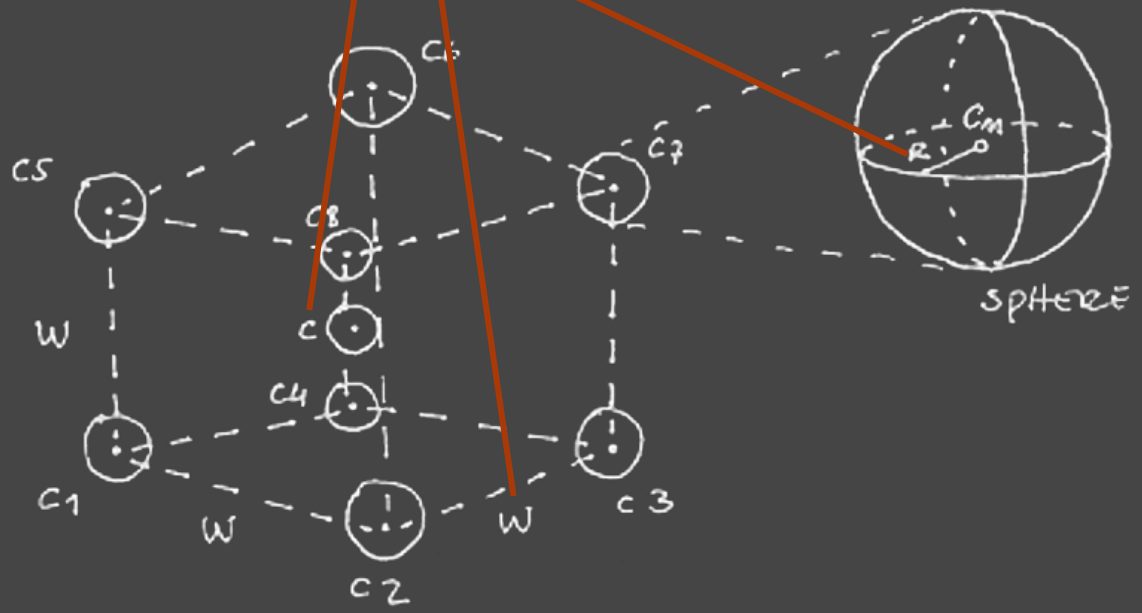



```
  c1 = c+xyz(-w,-w,-w)
  c2 = c+xyz(+w,-w,-w)
  c3 = c+xyz(+w,+w,-w)
  c4 = c+xyz(-w,+w,-w)
  c5 = c+xyz(-w,-w,+w)
  c6 = c+xyz(+w,-w,+w)
  c7 = c+xyz(+w,+w,+w)
  c8 = c+xyz(-w,+w,+w)
  for p in [c1,c2,c3,c4,c5,c6,c7,c8]:
    sphere(p, r)
```

Documentation

Program

```
def cubeSpheres(c, w, r):
```



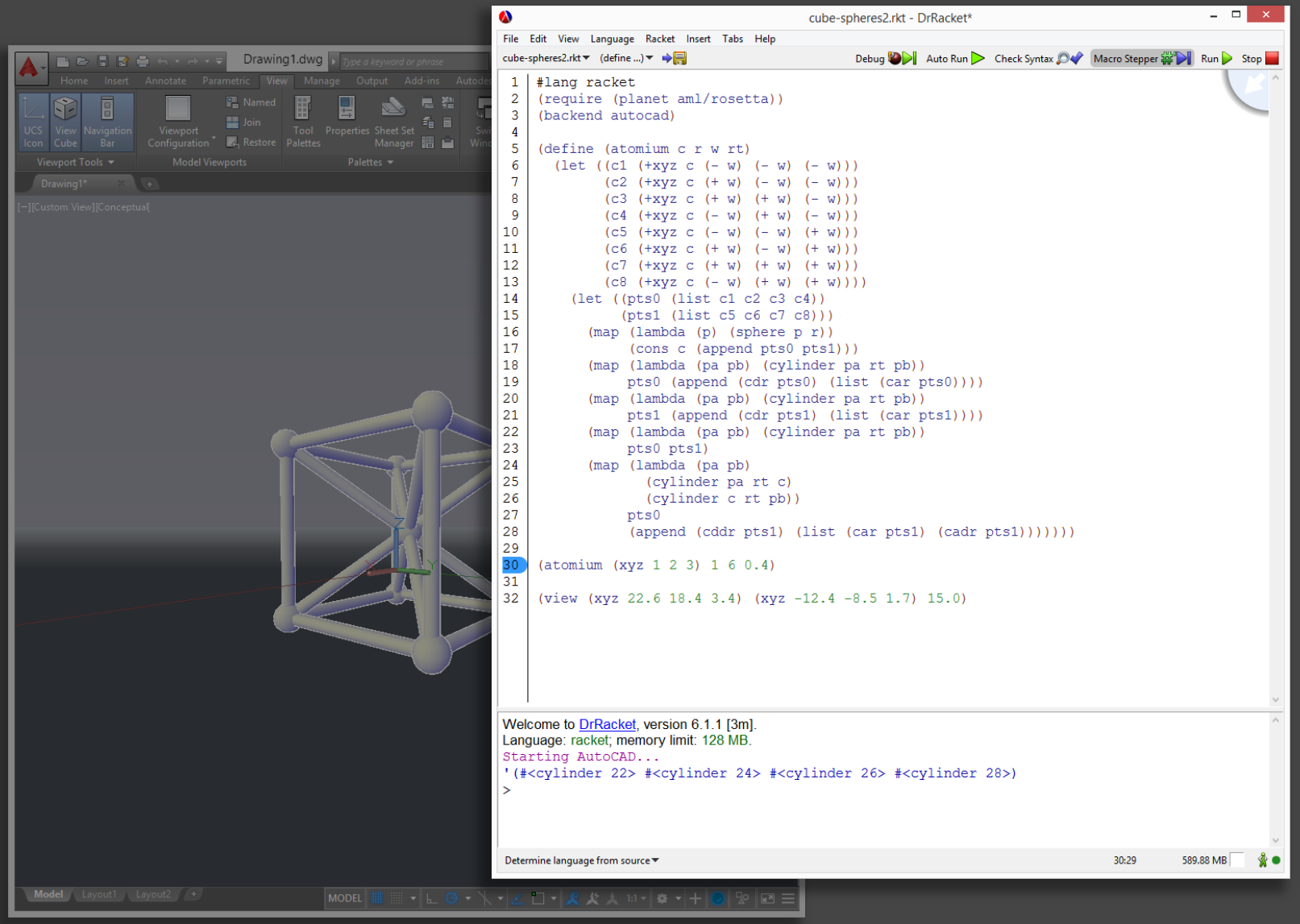
Documentation

```
  c1 = c+xyz(-w,-w,-w)
  c2 = c+xyz(+w,-w,-w)
  c3 = c+xyz(+w,+w,-w)
  c4 = c+xyz(-w,+w,-w)
  c5 = c+xyz(-w,-w,+w)
  c6 = c+xyz(+w,-w,+w)
  c7 = c+xyz(+w,+w,+w)
  c8 = c+xyz(-w,+w,+w)
  for p in [c1,c2,c3,c4,c5,c6,c7,c8]:
    sphere(p, r)
```

tickz.rkt (define ...) Debug Auto Run Check Syntax Macro Stepper Run Stop

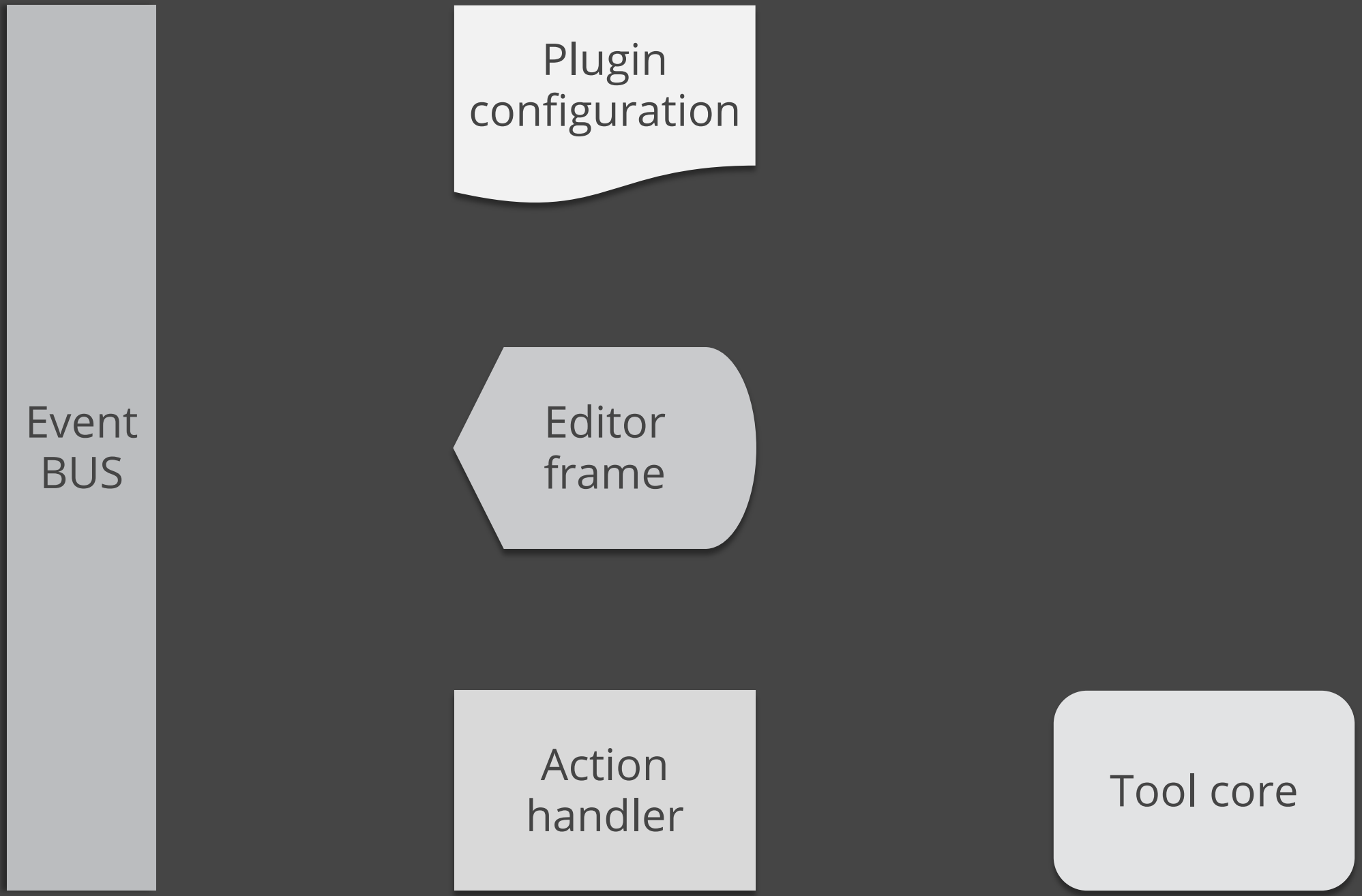
```
1 #lang racket
2 (require (planet aml/rosetta))
3 (require "sketch-correlation.rkt")
```

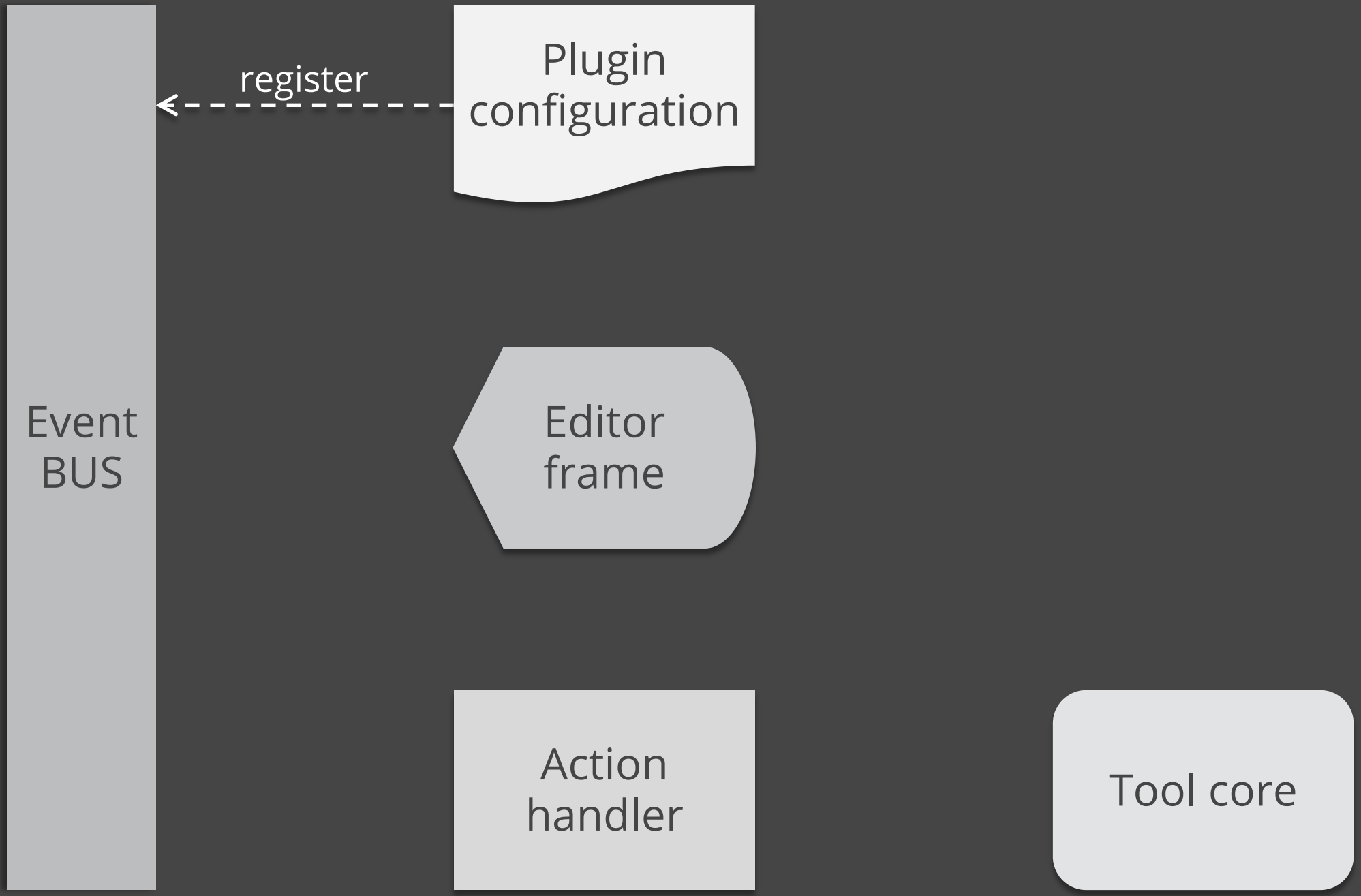
Creators need a medium connection

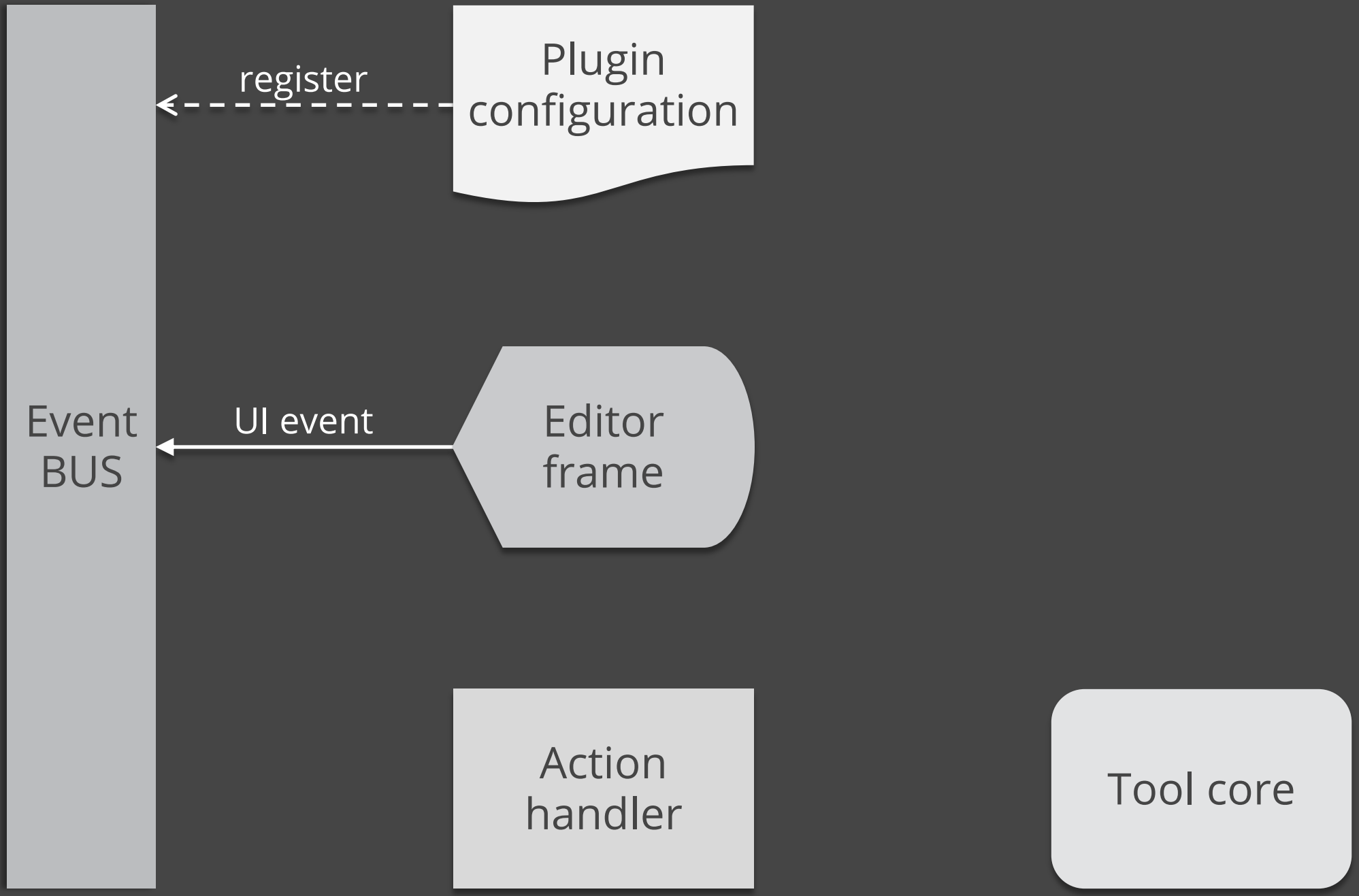


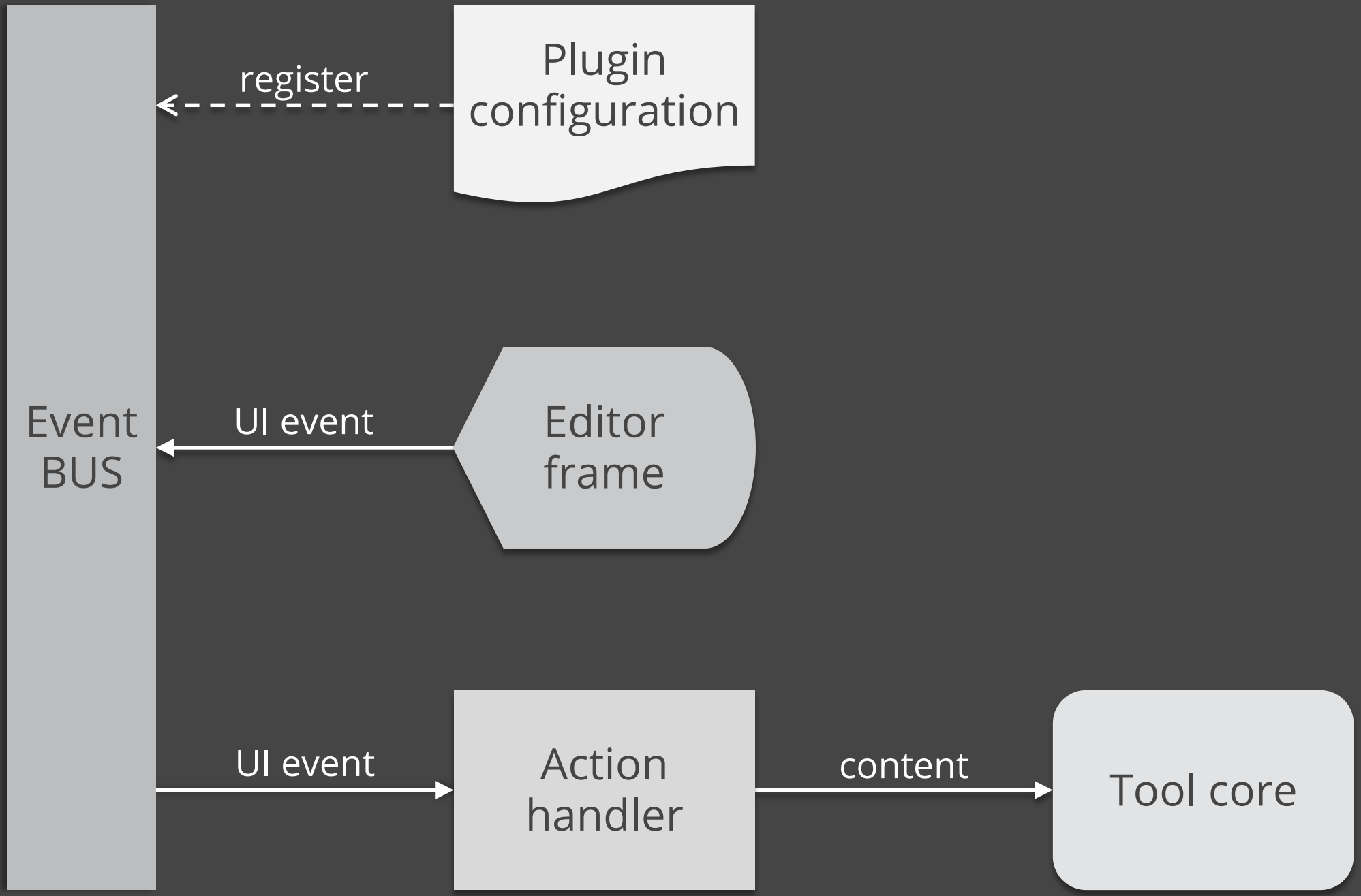
tickz.rkt (define ...) Debug Auto Run Check Syntax Macro Stepper Run Stop

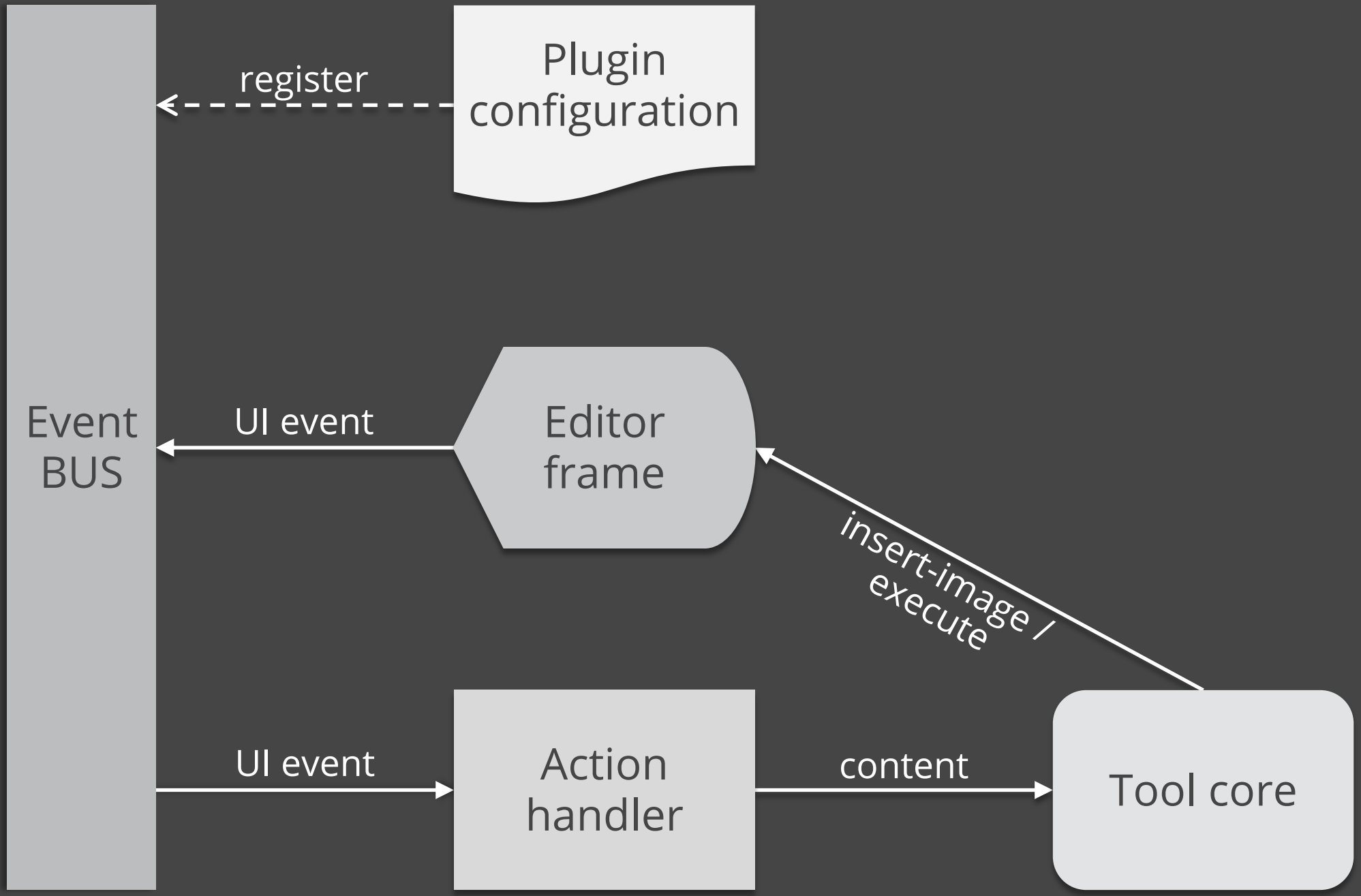
```
#lang racket
(require (planet aml/rosetta))
(require "tikzgen.rkt")
(backend tikz)

(define (arvore p c a da f)
  (let ((topo (+pol p c a)))
    (ramo p topo)
    (if (< c 2)
        (folha topo)
        (begin
          (arvore topo
                  (* c f)
                  (+ a da)
                  da f)
          (arvore topo
                  (* c f)
                  (- a da)
                  da f)))))

(define (ramo p topo)
  (line p topo))

(define (folha topo)
  (circle topo 0.2))



;render
(generateTikz)
```

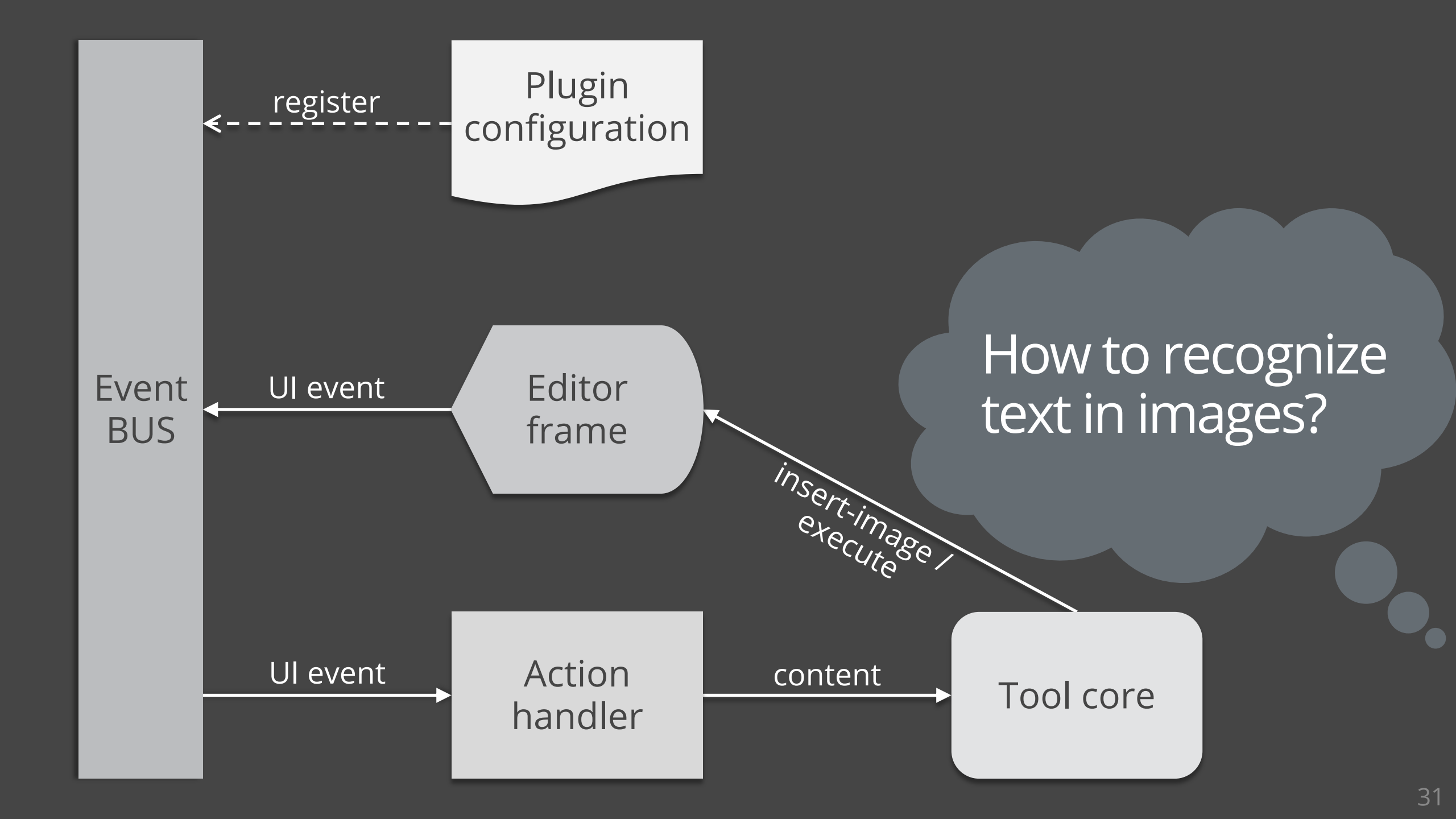

Event BUS
Plugin configuration
Editor frame
Action handler
Tool core

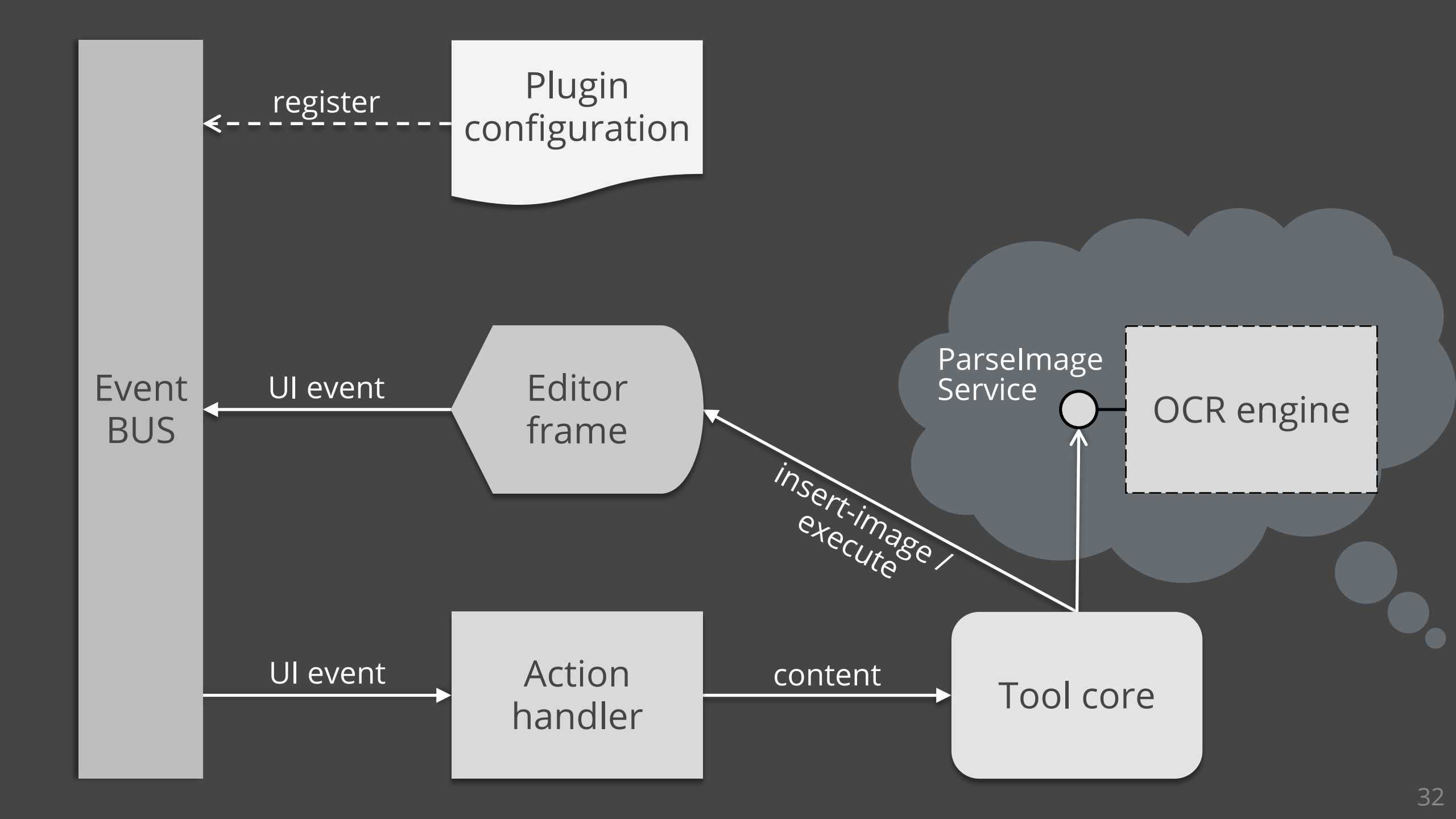

register
Plugin configuration
Event BUS
Editor frame
Action handler
Tool core


register
Plugin configuration
Event BUS
UI event
Editor frame
Action handler
Tool core


Plugin configuration
register
Event BUS
UI event
Editor frame
UI event
Action handler
content
Tool core


Event BUS
register
Plugin configuration
UI event
Editor frame
insert-image / execute
UI event
Action handler
content
Tool core


Event BUS
register
Plugin configuration
UI event
Editor frame
insert-image / execute
How to recognize text in images?
UI event
Action handler
content
Tool core

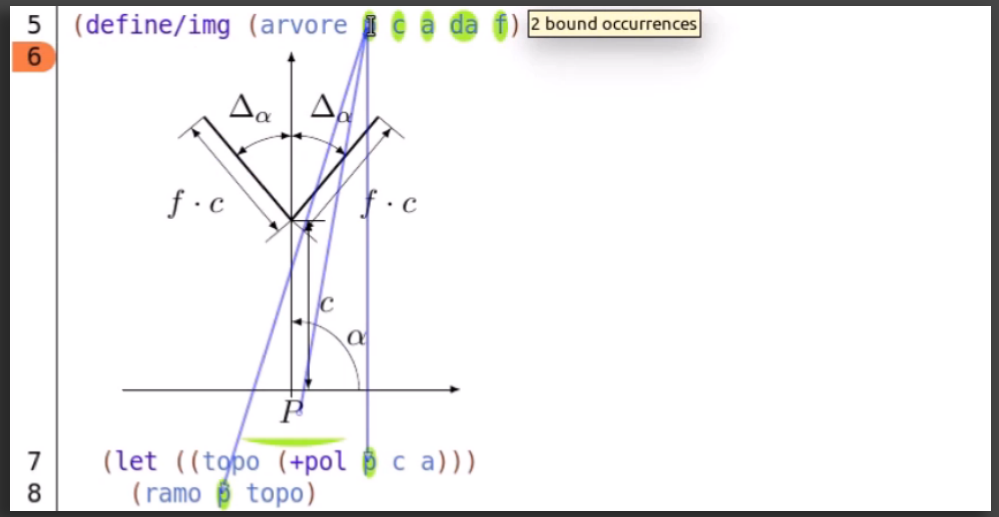

Event BUS
register
Plugin configuration
UI event
Editor frame
ParseImage Service
OCR engine
insert-image / execute
UI event
Action handler
content
Tool core

Evaluation

- Correctness
- Safety
- Performance
- Comparison

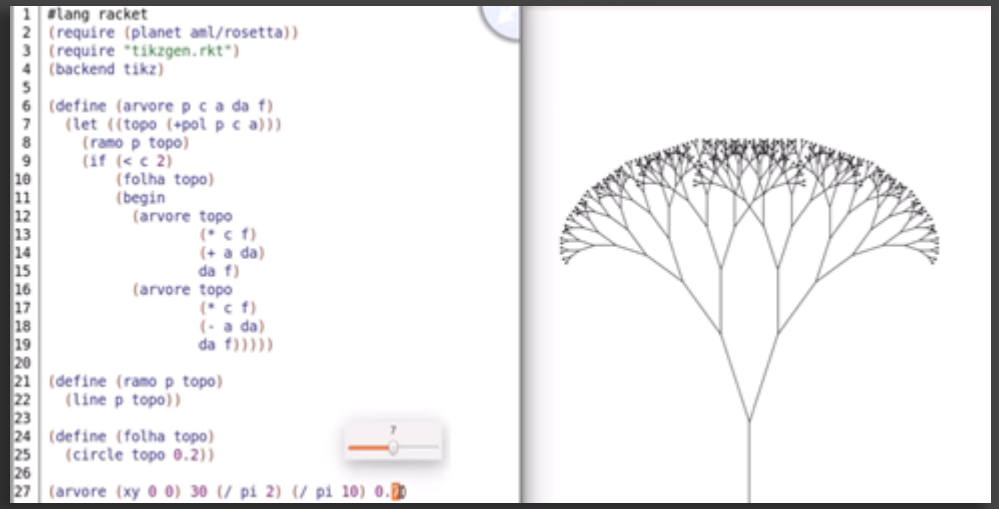
4

Conclusion



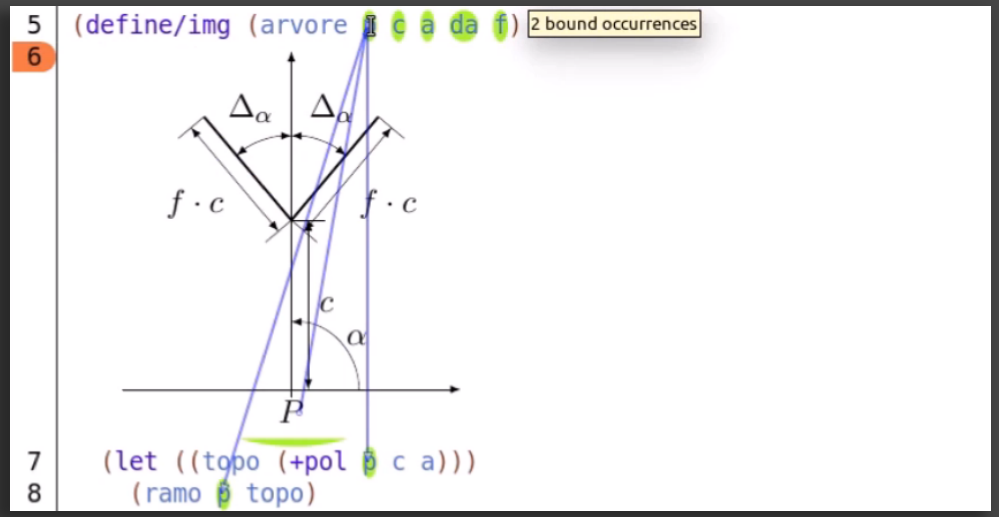
– Reduces the effort to understand the code

– GD is a particular case



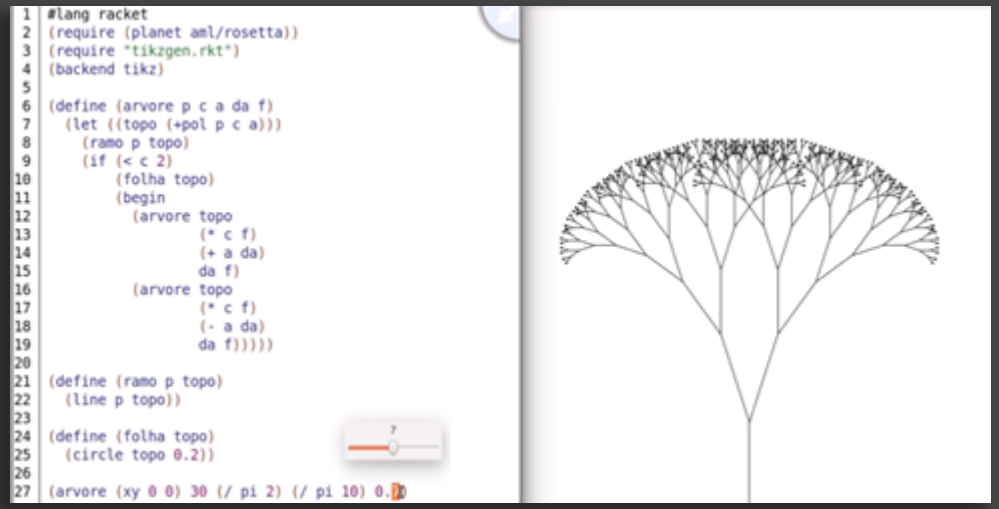
– Immediate feedback

– Performance



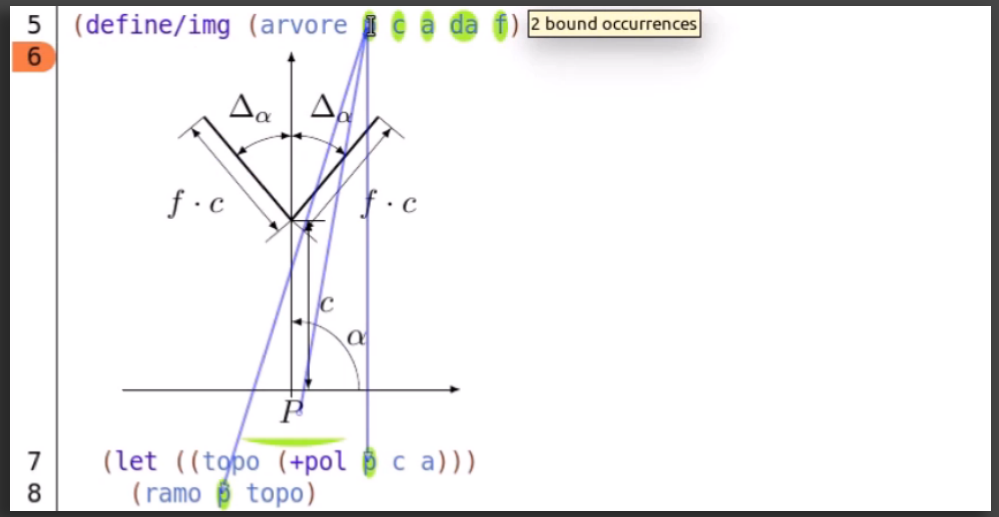
- Reduces the effort to understand the code
- GD is a particular case

- OCR Engine
- Support for image editing

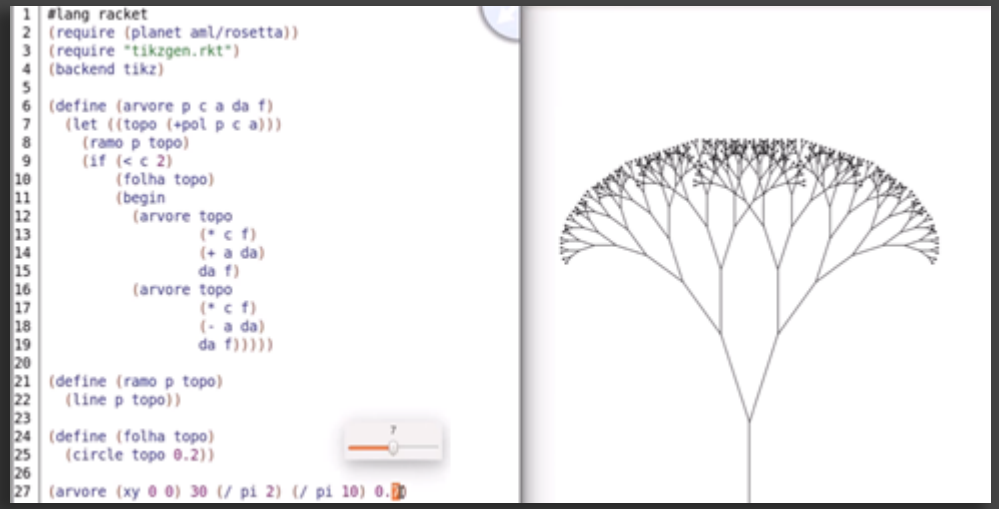


- Immediate feedback
- Performance

- Partial code execution
- Sandboxing



- Reduces the effort to understand the code
- GD is a particular case



- OCR Engine
- Support for image editing

- Immediate feedback
- Performance

- Partial code execution
- Sandboxing

Thank you. Questions?