Four

aspects

of making

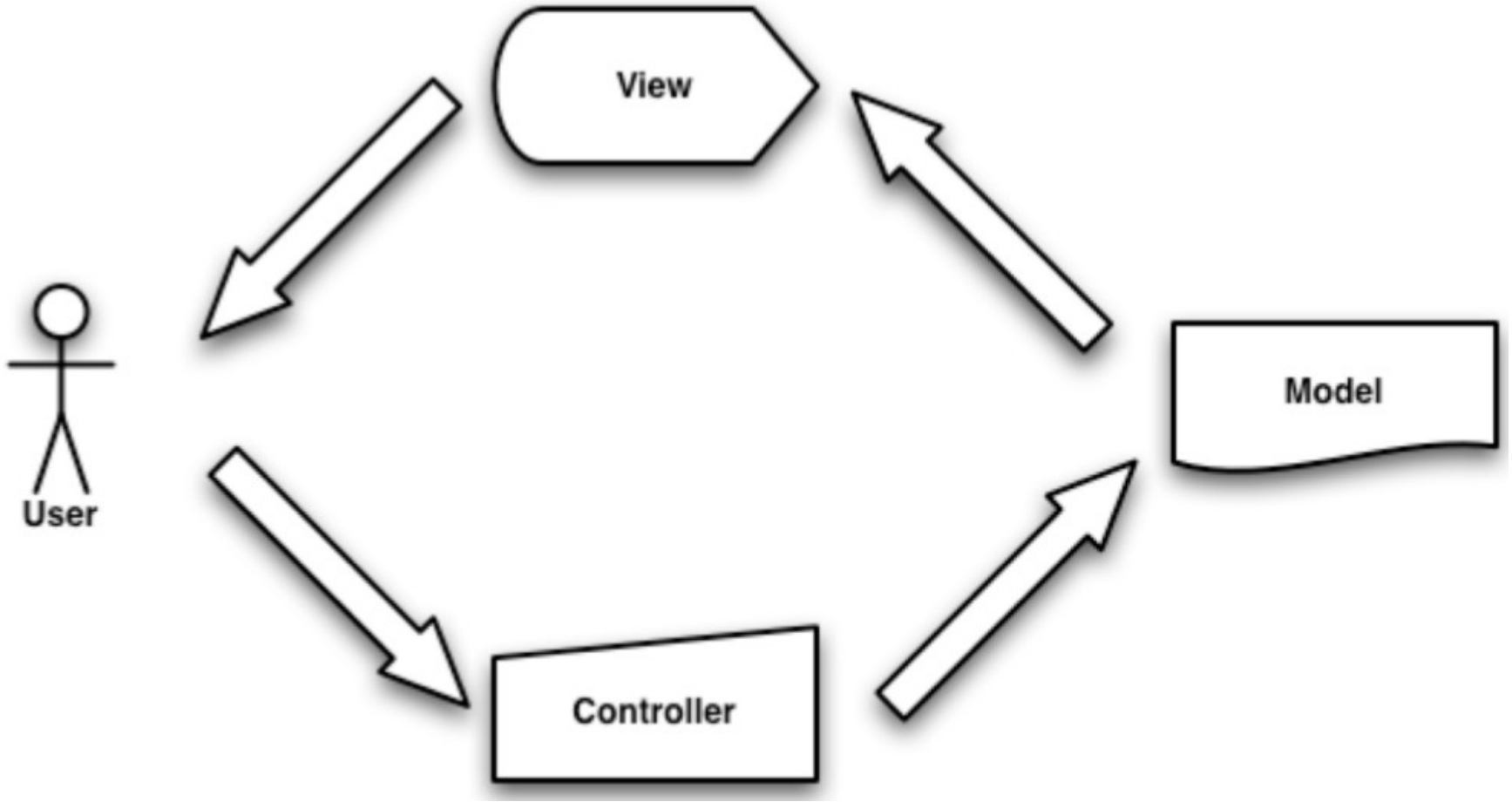
a Great consumer Operating System

- **UI**
- **Latency** Control
- Utilize **full** HW
- **Package** manager

UI · Basic Concepts

- **View**
- **Controller**
- **Model**

UI · Basic Concepts



UI · Basic Concepts · view

- **View**
  - Layout
  - Drawing
    - *theme engine*

UI · View · Layout

- Clutter is a good example...
- ClutterLayoutManager
  - ClutterBinLayout
  - ClutterBoxLayout
  - ClutterFixedLayout
  - ClutterFlowLayout
  - ClutterTableLayout

UI · View · Drawing

- cairo -- 2D vector drawing
- cogl -- wrap OpenGL with OOP
- ...

UI · Drawing · example

Drawing a button with cairo

UI · Drawing · example

main(int argc, char **argv):

```
GtkWidget *window, *drawing_area;
gtk_init(&argc, &argv);

window = gtk_window_new(GTK_WINDOW_TOPLEVEL);

drawing_area = drawing_area_new();

gtk_container_add(GTK_CONTAINER(window), clock);

g_signal_connect(window, "destroy",
                 G_CALLBACK(gtk_main_quit), NULL);
gtk_widget_show_all(window);
gtk_main();
```

UI · Drawing · example

drawing_area_new():

```
GtkWidget *wid = gtk_drawing_area_new();
gtk_widget_set_size_request(wid,
                            WIDTH, HEIGHT);
g_signal_connect(G_OBJECT(wid), "expose_event",
                 G_CALLBACK(draw_button), NULL);
return wid;
```

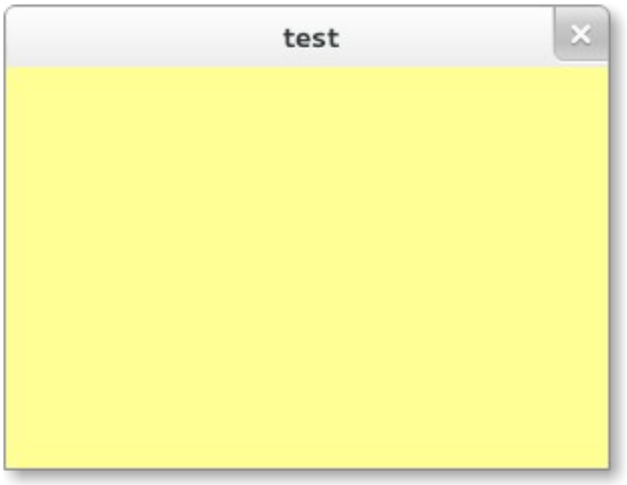
UI · Drawing · example

draw_button(GtkWidget *widget, ...:

```
GdkWindow *win = gtk_widget_get_window(widget);
cairo_t *cr = gdk_cairo_create(win);

GtkAllocation alloc;
gtk_widget_get_allocation(widget, &alloc);
cairo_translate (cr, alloc.x, alloc.y);

/* draw a backgroup color  */
cairo_set_source_rgb (cr, 1.0, 1.0, 0.588);
cairo_rectangle(cr, 0, 0, WIDTH, HEIGHT);
cairo_fill (cr);
```

UI · Drawing · example

What we got:



UI · Drawing · example

draw_button() cont:

```
#define BUTTON_X ((WIDTH - BUTTON_WIDTH)/2)
#define BUTTON_Y ((HEIGHT - BUTTON_HEIGHT)/2)

cairo_move_to(cr, BUTTON_X, BUTTON_Y);
cairo_rectangle(cr, BUTTON_X, BUTTON_Y,
                    BUTTON_WIDTH, BUTTON_HEIGHT);
```

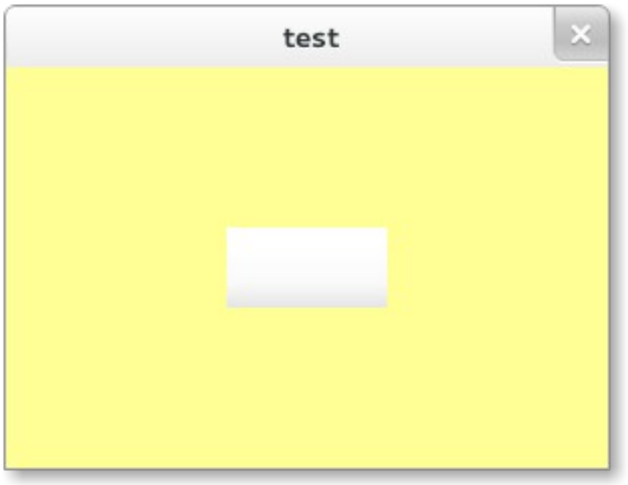
UI · Drawing · example

draw_button() cont -- set gradient

```
pattern = cairo_pattern_create_linear(
          BUTTON_X, BUTTON_Y,
          BUTTON_X, BUTTON_Y+BUTTON_HEIGHT);
cairo_pattern_add_color_stop_rgb (
          pattern, 0.0, 1.0, 1.0, 1.0);
cairo_pattern_add_color_stop_rgb (
          pattern, 0.7, 0.98, 0.98, 0.98);
cairo_pattern_add_color_stop_rgb (
          pattern, 1.0, 0.9, 0.9, 0.9);
cairo_set_source (cr, pattern);

cairo_fill (cr);
```

UI · Drawing · example

What we got:



UI · Drawing · example

draw_button() cont -- draw_label()

```
...
draw_label(cr, "确定");
cairo_destroy(cr);
return TRUE;
```

draw_label(cairo_t *cr, char *label)

```
cairo_save(cr);
cairo_set_source_rgb (cr, 0.0, 0.0, 0.0);
```

UI · Drawing · example

draw_label() cont:

```
PangoLayout *pl = pango_cairo_create_layout(cr);
PangoFontDescription *font_desc = \
  pango_font_description_from_string("Sans 14");

pango_layout_set_text(pl, label, -1);
pango_layout_get_pixel_size (pl,
                    &label_width, &label_height);

label_x = BUTTON_X+(BUTTON_WIDTH-label_width)/2;
label_y = BUTTON_Y+(BUTTON_HEIGHT-label_height)/2;
```

UI · Drawing · example

draw_label() cont:

```
cairo_move_to(cr, label_x, label_y);
pango_cairo_show_layout(cr, pl);

g_object_unref(pl);
pango_font_description_free(font_desc);
cairo_restore(cr);
```

UI · Drawing · example

what we got:

UI · View · theme engine

- ClutterEffect
- gtk-engines(GtkStyle), icon theme

UI · Basic Concepts · controller

- **Controller**
  - Event & Event dispatching
    - wrap around window system
    - clutter is a good example (again)
      - capture -> bubble

UI · Basic Concepts · model

- **Model**
  - Event callback (should be an obsolete fashion)
  - Property Model

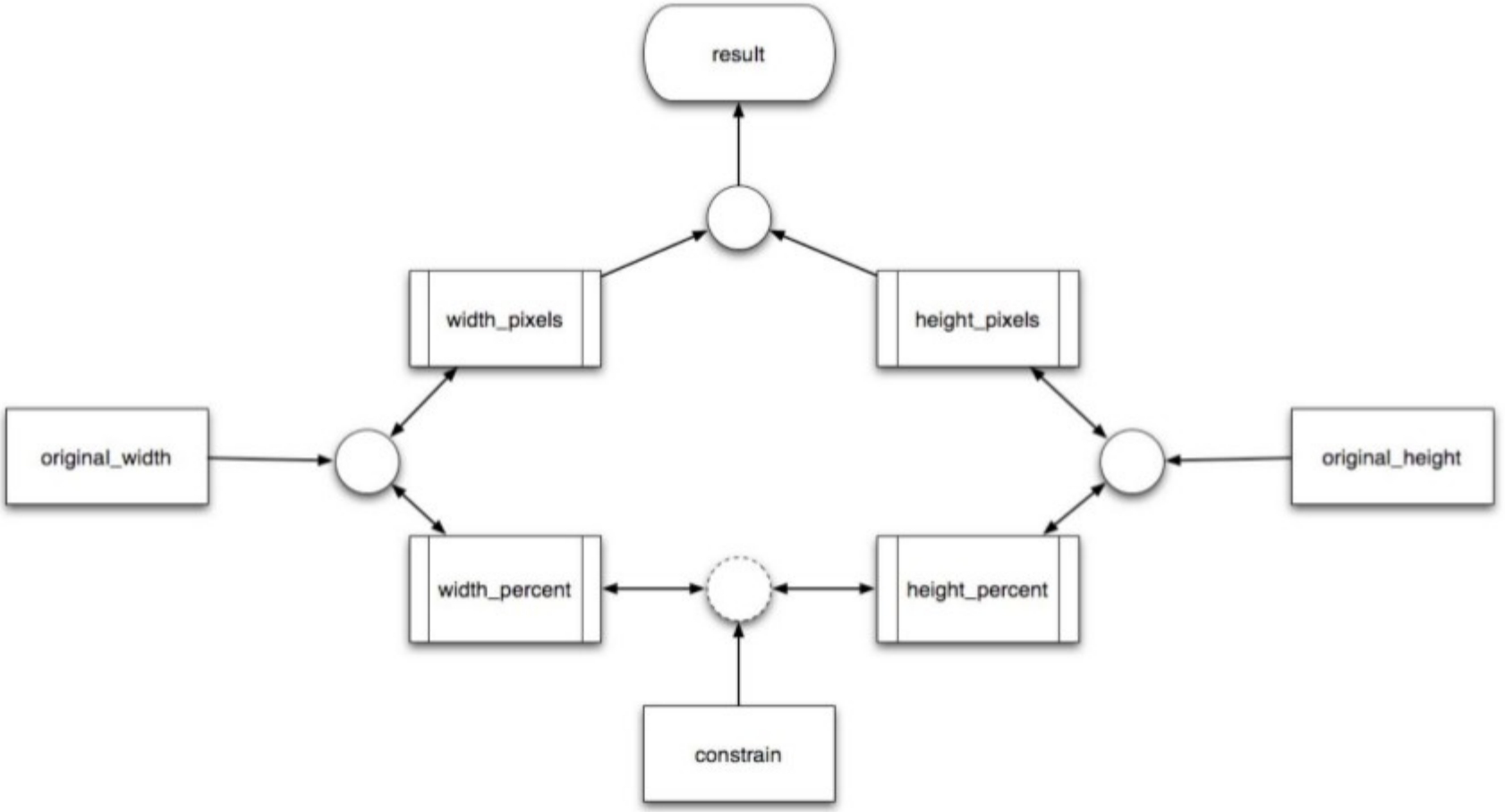
UI · model · Adam

- Property model declarative language
- Initial from adobe
  - ASL -- adobe source libraries
  - Open source, under MIT License
- Similar to a traditional spreadsheet

UI · Adam · example

Mini-Image Size Example

UI · Adam · example

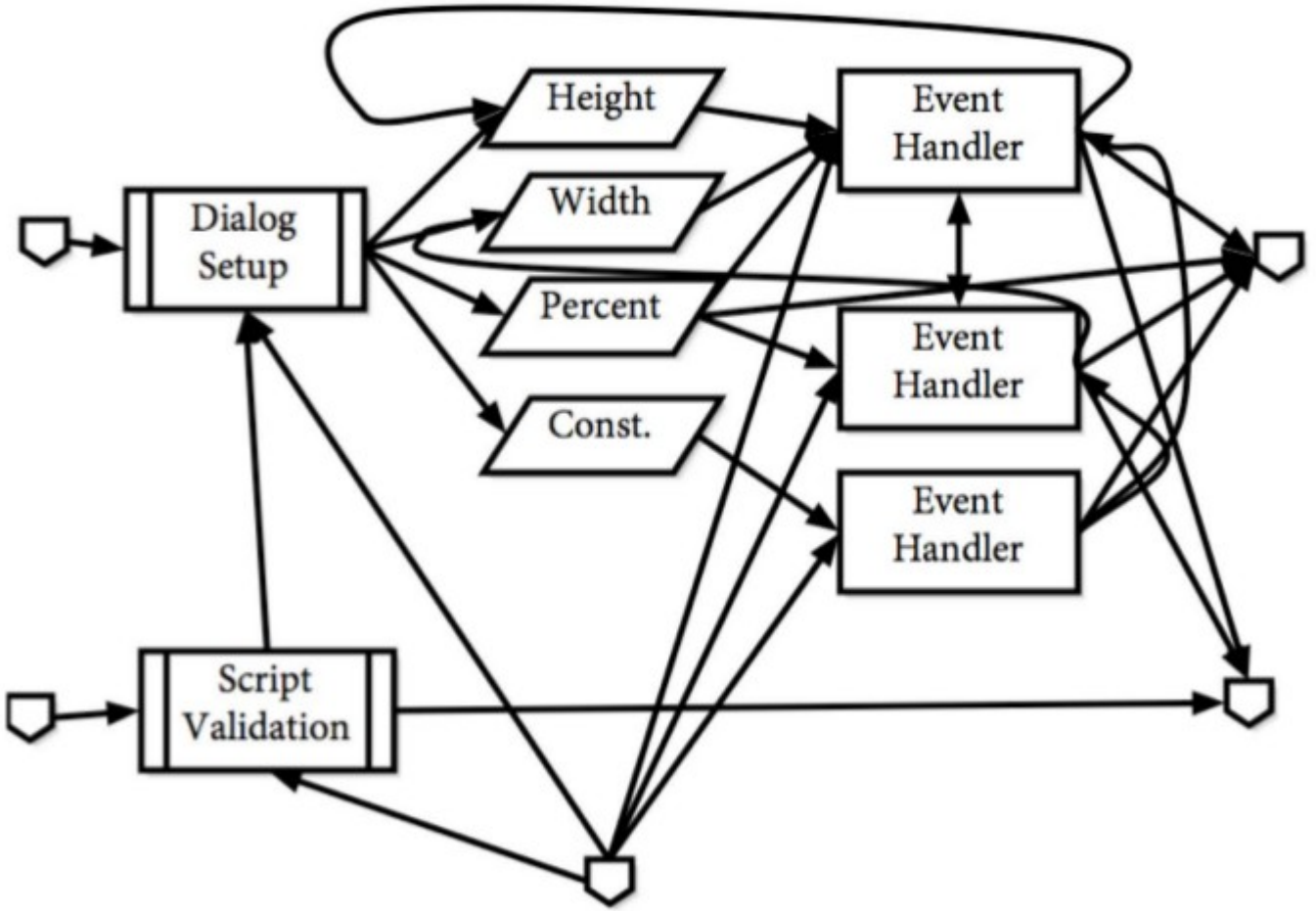


UI · Adam · example

```
sheet mini_image_size
{
input:
  original_width    : 2304;
  original_height   : 1296;
interface:
  constrain     : true;
  width_pixels  : original_width <== round(width_pixels);
  height_pixels : original_height<== round(height_pixels);
  width_percent;
  height_percent;
logic:
  relate {
    width_pixels <== round(width_percent * original_width / 100);
    width_percent<== width_pixels * 100 / original_width;
  }
  relate {
    height_pixels <== round(height_percent * original_height / 100);
    height_percent<== height_pixels * 100 / original_height;
  }
  when (constrain) relate {
    width_percent <== height_percent;
    height_percent<== width_percent;
  }
output:
  result <== { height: height_pixels, width: width_pixels };
}
```

UI · Adam · example

Event Flow in a Simple User Interface



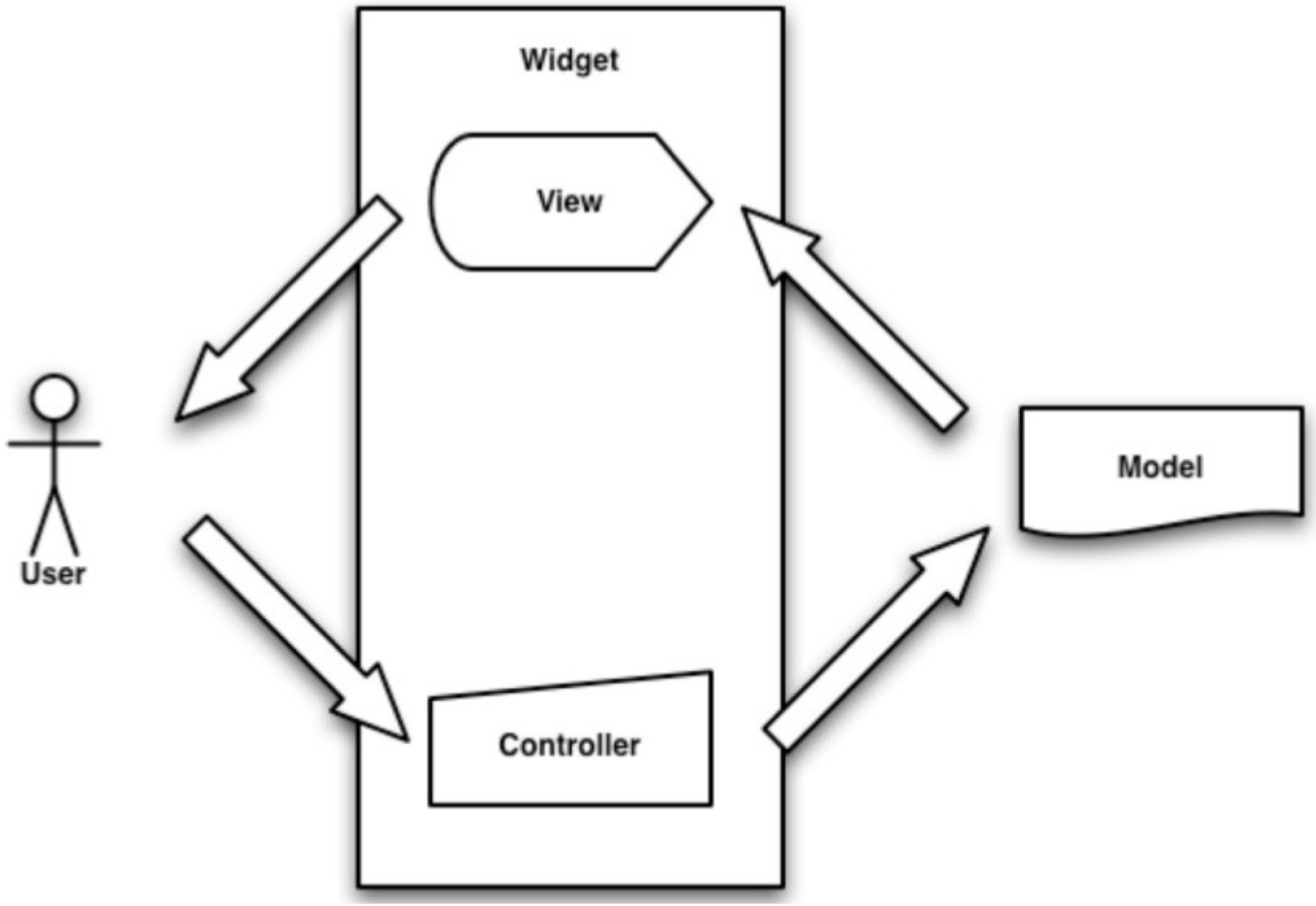
UI · Adam · discuss

- Add 'reason' to indicate which input causes an unacceptable result
- Help to mask some core functions on a specific platform
  - Not touch the core
  - Build a property model
  - Set variables to mask some core functions, related functions/UI will be masked together

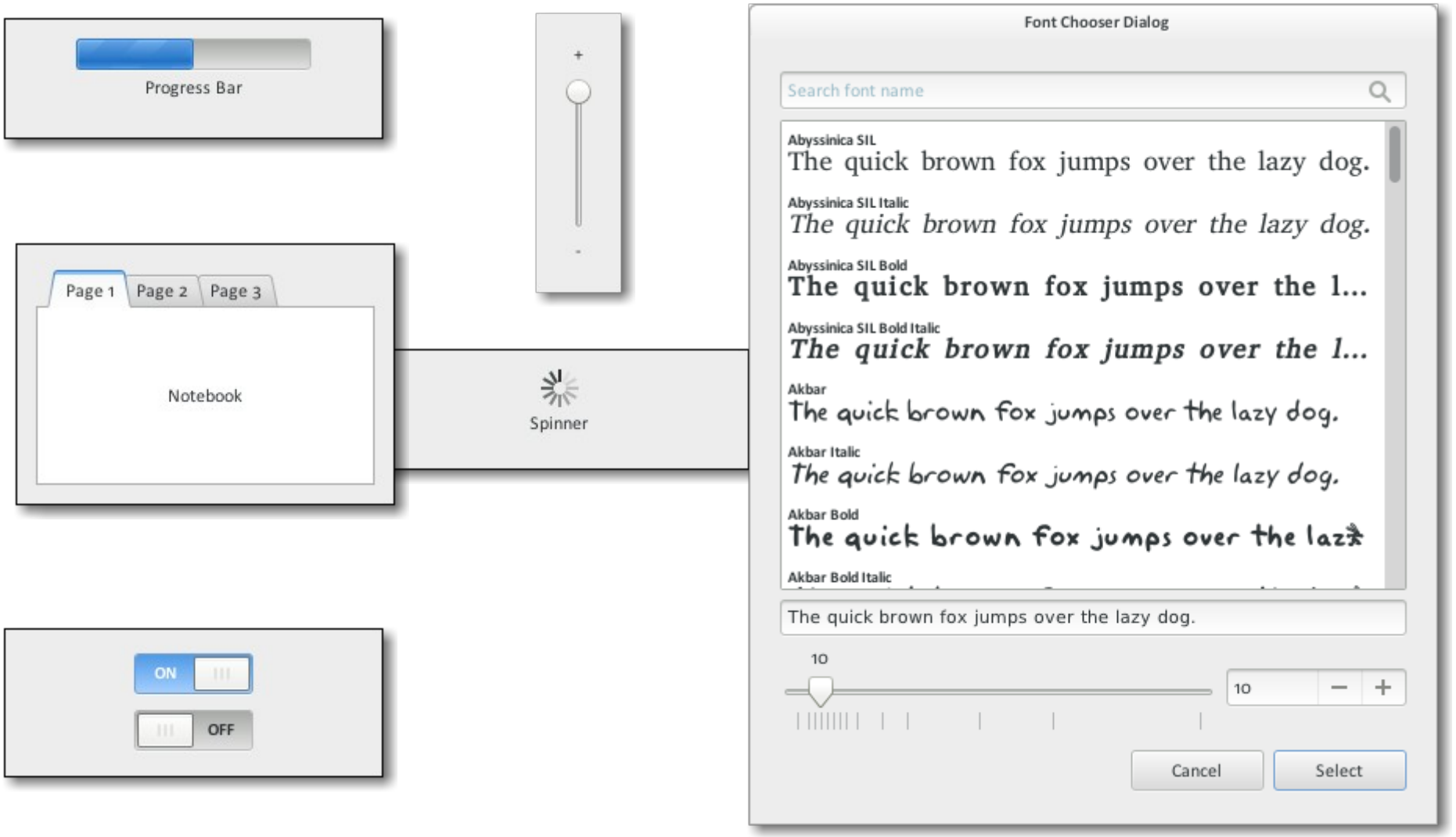
UI · Concepts · more

- **Widget**
- **Animation**
  - *physics engine*

UI · Concepts · widget



UI · Concepts · widget



UI · Concepts · animation

- Animations are eye-catching
- Take clutter as a (good) example

UI · Concepts · animation

```
{
  "object"  : "rectangle",
  "name"    : "x",
  "ease-in" : true,
  "keys"    : [
    [ 0.0, "linear", 0.0 ],
    [ 0.1, "easeInCubic", 150.0 ],
    [ 0.8, "linear", 150.0 ],
    [ 1.0, "easeInCubic", 0.0 ]
  ]
}
```

- Timeline, Key frame
- Alpha function

UI · Concepts · animation

- Physical-based animation
  - clutter-box2d
  - clutter-bullet

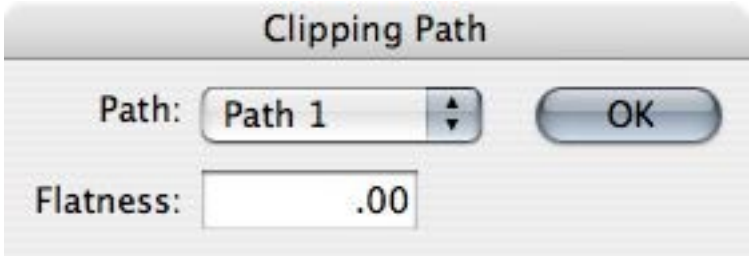
UI · Basic Principles

- Separate UI from core
  - Sharable
  - Easy to modify
- Using Specific Language for
  - Layout, Drawing, Widget, theme
  - Model
  - Animation

UI · Language

- **Layout & widget**
  - Gtk builder(XML), Clutter script(JSON), Eve ...
- **Drawing**
  - GLSL, svg, Image?
- **Theme**
  - CSS
- **Model**
  - Adam
- **Animation**
  - Clutter script

UI · Lang · Eve · example




```
layout clipping_path
{
    view dialog(name: "Clipping Path")
    {
        column(child_horizontal: align_fill)
        {
            popup(name: "Path:", bind: @path, items:
            [
                { name: "None", value: empty },
                { name: "Path 1", value: 1 },
                { name: "Path 2", value: 2 }
            ]);
            edit_number(name: "Flatness:", digits: 9, bind:
@flatness);
        }
        button(name: "OK", default: true, bind: @result);
    }
}
```

Latency Control

- Latency analysis
- Preserve Resource
- Async + MT
- Binary config files

Latency · analysis

- The operation is damn slow... what is the underlying system doing ?
  - Which points were passed? And the values of variables on each point.
    - Calling path
  - Check events, exceptions...

Latency · analysis

- Preferred ways provided by kernel
  - perf tool, ftrace
  - Based on tracepoint, kprobe and uprobe

Latency · ftrace · example

1. **Mount debugfs and enable ftrace**

```
# mount -t debugfs none /sys/kernel/debug
# echo 1 > /proc/sys/kernel/ftrace_enabled
# cd /sys/kernel/debug/tracing
```

2. **Function tracer for init**

```
# echo 1 > set_ftrace_pid
# echo function > current_tracer
```

3. **Cat result**

```
# cat trace
```

Latency · ftrace · example

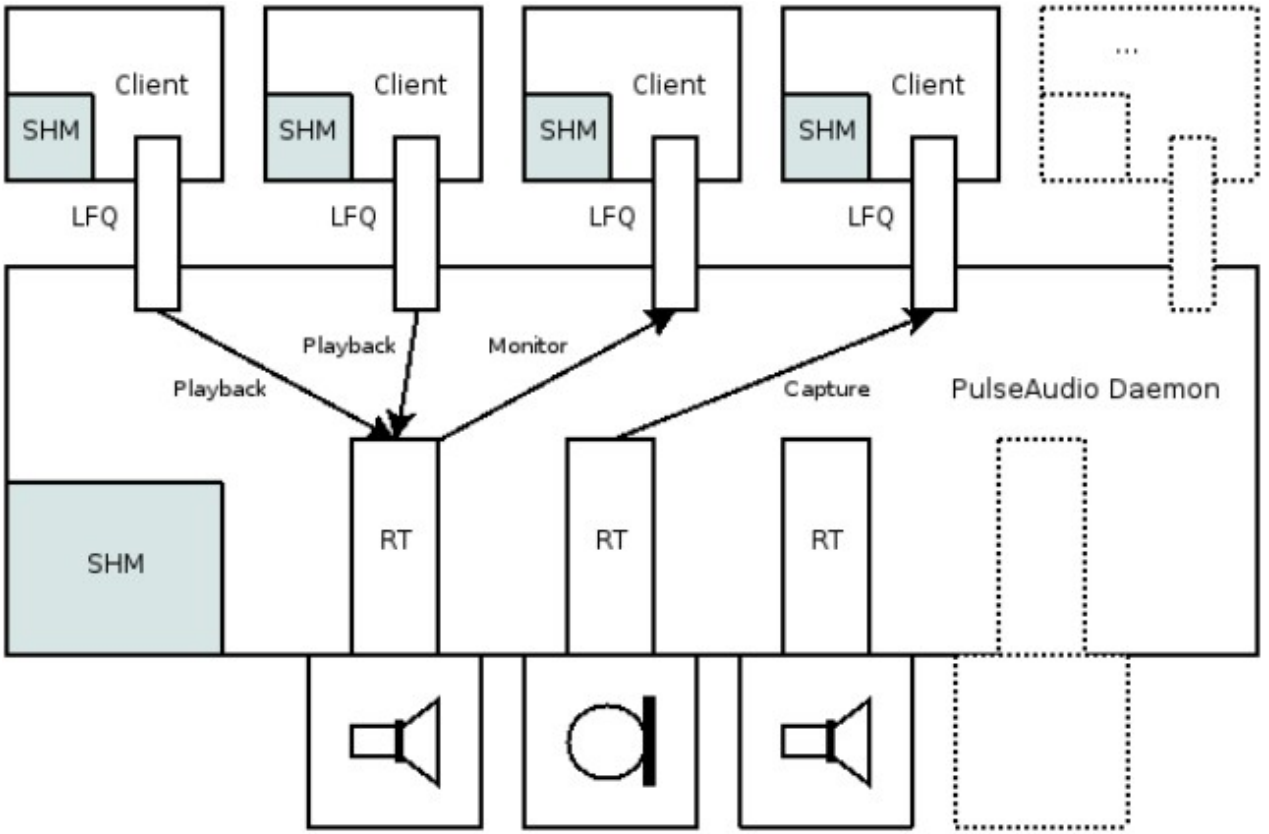
```
# tracer: function
#
# entries-in-buffer/entries-written: 837/837   #P:2
#
#                                _-----=> irqs-off
#                               / _----=> need-resched
#                              | / _---=> hardirq/softirq
#                              || / _--=> preempt-depth
#                              ||| /     delay
#           TASK-PID   CPU#   ||||    TIMESTAMP  FUNCTION
#              | |       |    ||||       |         |
           <...>-1     [000] d... 32602.717293:
finish_task_switch <-__schedule
           <...>-1     [000] .... 32602.717296:
_raw_spin_lock_irqsave <-sys_epoll_wait
           <...>-1     [000] d... 32602.717297:
_raw_spin_unlock_irqrestore <-sys_epoll_wait
           <...>-1     [000] .... 32602.717297:
ep_scan_ready_list.isra.7 <-sys_epoll_wait
           <...>-1     [000] .... 32602.717298: mutex_lock <-
ep_scan_ready_list.isra.7
          ...
```

Latency · Preserve Resource

- RT thread
- Memory pool & lock
- High IO prio

Latency · Preserve Resource

Example: how does PA employ RT?



LFQ = Lock-Free Queue; SHM = Shared Memory Segment; RT = Realtime Thread

Latency · Async+MT

- libdispatch
  - e.g.
    - Initiate IPC, IO simultaneously
    - Accessing net without blocking main thread

Latency · Binary config file

- Binary format of XML...
- Cache: one whole file instead of many small files

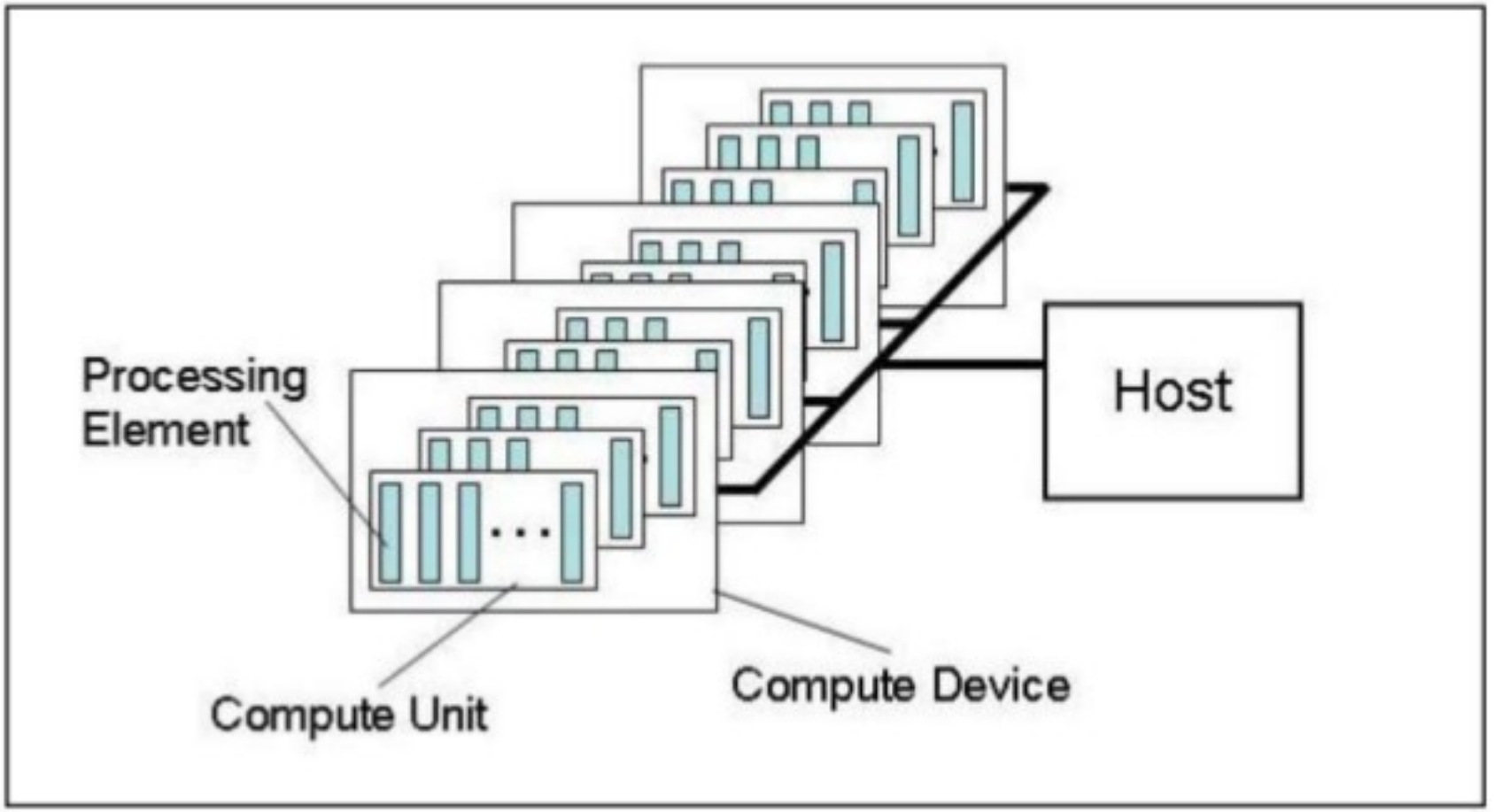
Utilize HW

- MultiCore
  - libdispatch
  - OpenCL
- Heterogeneous Computing
  - OpenCL

Utilize HW · Libdispatch

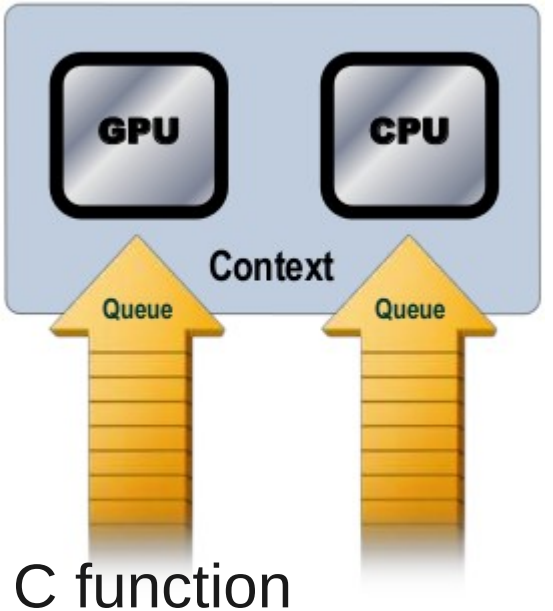
```
__block double sum = 0;

dispatch_queue_t q_default = \
dispatch_get_global_queue(0, 0);

dispatch_queue_t q_sum = \
dispatch_queue_create("com.example.sum", NULL);

#define COUNT 128
dispatch_apply(COUNT, q_default,
 ^(size_t i){
     double x = complex_calculation(i);
     dispatch_async(q_sum, ^{ sum += x; });
 });

dispatch_release(q_sum);
```

Utilize HW · OpenCL



Utilize HW · OpenCL

- **OpenCL application runs on a host which submits work to the compute devices**
  - **Context**: The environment within which work-items executes ... includes devices and their memories and command queues
  - **Program**: Collection of kernels and other functions (Analogous to a dynamic library)
  - **Kernel**: the code for a work item. Basically a C function

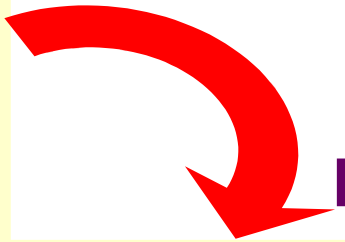


- **Work item**: the basic unit of work on an OpenCL device
- **Applications queue kernel execution**
  - Executed in-order or out-of-order

Utilize HW · OpenCL

```
void
trad_mul(int n,
         const float *a,
         const float *b,
         float *c)
{
  int i;
  for (i=0; i<n; i++)
    c[i] = a[i] * b[i];
}
```

Traditional loops



Data Parallel OpenCL

```
kernel void
dp_mul(global const float *a,
       global const float *b,
       global float *c)
{
  int id = get_global_id(0);
  c[id] = a[id] * b[id];
}// execute over
 // “n” work-items
```

Package Manager

- Essential of a Distro
- SW static resources management
  - Rule files to constrain execution, aka dynamic resources management
- Maintain relationships among packages
- SW Distribution/Update management

Package Manager · Turtle

- A package manager in concept
- Static resources of SW in Dir boundary
  - Static resources are readonly
  - Encapsulation
  - Easy to employ path-based AC
  - Sutiable to install to SSD/mount ro

Package Manager · Turtle

- Package Activation
  - Register well-known entries in public NS, e.g.
    - bin, library, .desktop, .service, etc
    - Plugin extension dir
- Continuous upgrade
  - vs WebApp:
    - More secure -- all files distributed through PM
    - Consistency -- no half rendered page!

Package Manager · Turtle

- Constrain execution, aka dynamic resources management
  - PM/Launcher load rule files to system
  - How to deal with tmp, log, cache? per-instance, session, user? global?
  - How to deal with data, user data, config, user config?

Package Manager · Turtle

- Package Name == Source Name
- Package = Meta + SW Binary
  - Update a package:
    - update meta
    - update SW Binary
- SW Binary = Source + config/build args + build env

Package Manager · Turtle

- Section of a package = subset of a SW binary
- Variant of a package =
  - the same source +
  - a different config/build arg and/or build env
- Package Meta = Seed + Manifest
- Seed is a snippet of package DB
- Manifest contains file fingerprints of a package section

Package Manager · Turtle

Seed

- Package Name
- Package Version
- Seed Version
- 枚举变种 (Variant) ，及其描述
- Sections ...
  - Variants
    - 描述
    - 依赖关系
    - Manifest，其文件大小及 Hash 码
- 签名

Manifest

- 标识： PackageName[Version].Section[Variant]
- 包含的文件、其大小及其 Hash 码

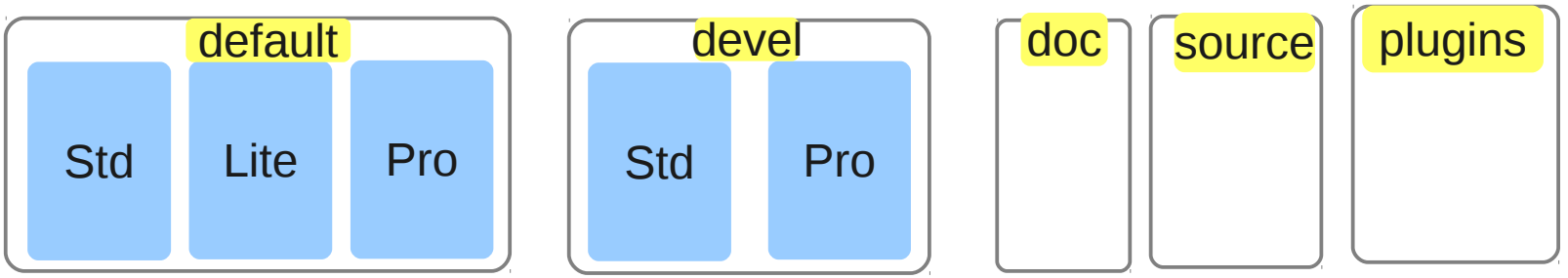
Package Manager · Turtle

Name: Foo
Ver: 1.0
SeedVer: 0.1

Vars: Std **Lite** Pro

- Std: 标准版
- Lite: 轻量版
- Pro: 专业版

Sections:



The END